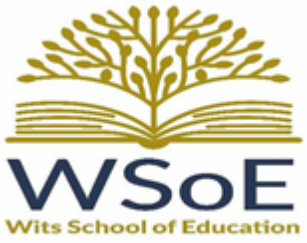


Enhancing the teaching and learning of programming functions and procedures using legitimate code theory

By

Nothando Truely Wadi

(1741809)

Research submitted in fulfillment of the Requirement for the Master of Education degree

In the Faculty of Humanities

EDUCATIONAL INFORMATION AND ENGINEERING TECHNOLOGY (EDIET) DIVISON

Wits School of Education

Supervisor: Dr. Alton Dewa

Date: February 2023

Copyright

Acknowledgement

Without God, this research would not have been possible. He has guided and strengthened me and my supervisor throughout the process and toward the completion of this research. I would also like to thank my supervisor (Dr. Alton Dewa), who guided me and provided constant support throughout the research process, who has sacrificed his time, efforts, and commitment to ensure that I receive all the guidance and assistance need to make this research a success. I would also like to thank my wonderful mother for all the motivation she has provided me, she has always pushed me to keep going even at my most drained and stressed moments, she has provided financial assistance for all traveling during the data collection process, thank you so much mom, you are appreciated in all that you do. I would also like to thank my colleagues for the assistance provided throughout. Most of all I would like to dedicate to my past Dad who was a teacher and had so much passion for education unfortunately he didn't get the chance to witness me graduate and peruse postgraduate. I hope you are proud of how far we have come, Dad. I would also like to thank the National Research Foundation for financial assistance (NRF) this research wouldn't have been possible had it were not for foundations assistance.

Abstract

Learners in high schools find programming difficult especially the concepts of functions and procedures since these concepts are covered at the very beginning of the subject, the challenge begins in Grade 10 and escalates to further grades and tertiary education which results to learners drop out in the specific subject and opting for more flexibles subject such as Computer Applications Technology (CAT). The Research showed that factors that contribute to this challenge ranges from inappropriate teaching methods used by educators to inability to follow appropriate or correct syntax when coding by learners. In addition, the research revealed that there is confusion of differentiating between functions and procedures by learners due to the technicality of the language used in learner and teacher resources such as textbooks. The main contributing reason seems to be a lack of teacher development in programming to help guide and equip teachers with skills that may provide them with knowledge to teach programming in the information technology topic at a level that is more understandable to learners. This study aims to unpack the mingling of functions and procedures settings and repack distinct settings by going from simpler meanings [concrete] to complex meanings [abstract] using the semantic wave construct of the Legitimation Code Theory (LCT). Semi-structured interviews were conducted with participants (Information Technology teachers) teaching programming languages of either Java or Delphi in grade 10. Four participants were interviewed and responded to all questions asked. The results showed that improper use of code syntax results in failure to execute running codes and the need for IT teacher's professional growth. The research indicated that teacher's poor selection of teaching methodologies as well as the dominance of technical language used when teaching learners as well as contained in learner's learning resources such as textbooks are the two main reasons learners mingle the notion of functions and procedures. Thus, there is a need to uncover what the suitable teaching methodologies are for teaching programming concepts such as functions and procedures. In addition, extensive use of the LCT to unpack not only functions and procedures but all complex concepts in programming, ranging from the way teachers structure their teaching instructions to structuring activity question, assignments, or tests. The study recommended that IT educators employ the idea of LCT's semantic waves to break down technical language to a level that learners could understand without losing the concepts' original meaning.

Key words: Information Technology (IT), Functions, Procedures, Legitimate Code Theory, Semantic Waves.

Contents

List of acronyms

IT	Information Technology
LCT	Legitimate Code Theory
CAPS	Curriculum Assessment Policy Statements

Chapter one: Background of the study

1.1. Introduction

This chapter starts with an introduction to the research and explains the research questions that result from the issue that will be covered in this study. The study's aim, which is to provide answers to the research questions, is outlined in this chapter. Both the purpose and the appeal of the study have been emphasized. The conclusion of this chapter includes an explanation of how the study was organized as well as a final observation.

1.2. Background overview

Information Technology (IT) and Computer Applications Technology (CAT) are studies available in South African schools from grades 10 to 12. (Jacobs & Sewry, 2010). The two subjects differ, in that, CAT specializes in using ICTs to address issues with pattern recognition, organization, and interaction of information (Koorsse, Calitz & Cilliers, 2010). However, IT equips students with the knowledge and abilities to comprehend intricate ICT equipment and software fundamentals, apply digitization to issues, and use coding to build, create, and improve processes to address these issues (Jacobs & Sewry, 2010).

Programming is a language for computers, since computers speak in codes, humans require computers to carry out certain activities through programming (Blackwell, 2002). South African schools doing programming are mostly focusing on either Java or Delphi programming languages (Idrees & Aslam, 2022). Meaningful learning involves more than simply recalling the definition of a particular concept, learners need to comprehend the explanation, descriptions as well as examples of concepts. Additionally, learners need to hold the capability to apply skills learned through theoretical information in practical contexts. When it comes to information technology, learners often excel in recalling the definition of concepts, including that function and procedure. However, learners prove to perform poorly upon implementing the concepts practically. Learners mingle notions of functions and procedures, resulting in failing to implement running codes. Programming codes are strict in using their languages, once the syntax is not followed or if any of the syntax regulations are broken the code will not run (Idrees & Aslam, 2022).

According to Department for Education (2014), the National Curriculum Statement for Information Technology in Grades 10-12 expresses the information, abilities, and values that South African students should learn. The goal of this curriculum is for learners to attain and

develop skill sets that are relevant to their own lives, in this way, the curriculum encourages learning in contextual factors while remaining mindful of international implications (Department for Education, 2014). The curriculum strives to provide learners with the knowledge, skills, and values necessary for self-fulfilment and meaningful involvement in society as citizens of a free democracy, regardless of their socioeconomic background, ethnicity, sexuality, athletic strength, or mental ability (Department for Education, 2014). This implies the fact that learner should be permitted to access to an equal education that enables them to exercise their fullest potential to acquiring relevant knowledge and skills necessary and beyond the mastery of the subject. It also intends to make higher education more accessible (Department for Education, 2014). The following concepts underpin the National Curriculum Statement for Grades R-12: Active and critical learning: rather than rote and uncritical learning of given truths, encourage an active and critical attitude to learn (Department for Education, 2014). High knowledge and skills: Minimum levels of knowledge and skills to be obtained at each grade are established, and all topics have high, attainable standards (Department for Education, 2014). Another premise is legitimacy, quality, and effectiveness: delivering a quality, breadth, and depth of education equivalent to that of other countries (Department for Education, 2014). Regardless of the goals of the national curriculum statements, which are to provide more effective, more accessible, and understandable education for information technology learners. Grade 10 Information Technology students continue to struggle with learning and understanding functions and procedures (Department for Education, 2014). The very moment that there is an increased number of learners mingling functions and procedures and most concepts in Information Technology in grade 10, an inequality that hinders learners' ability to exercise their right to access an equal education to their fullest potential. With such a case learners will encounter exclusion in participation in active and critical learning as well as competing with global education standards of programming. This will further affect their ability to use the skills learned in Information Technology in the work environment, the school, as well as the tertiary institution, will produce a large percentage of incompetence programmers if not encounter a decrement in the production of coders or programmers.

1.3. Problem Statement

In programming in Grade 10 classes, there is generally a lot of confusion and misunderstanding about teaching functions and procedures in programming in grade I0. Learners appear to be mingling their notions about when to utilize the function and when to apply the procedures (Idrees & Aslam, 2022). According to He, Zhang, and Li (2021), inexperienced programmers only have a cursory understanding of programming, they lack comprehensive cognitive structures, fail to implement basic elements, and use broad sense problem-solving techniques instead of programming or problem-specific ones. Understanding functions and procedures, which form the foundation of programming, is the main barrier to learning programming (He, Zhang & Li, 2021). The students in Grade 10 are brand-new to programming (since programming starts at grade 10 in the South Africa syllabus), making them unskilled programmers who need practical training to understand the principles (He, Zhang & Li, 2021). A concept of the problem domain, such as Functions and procedures, is also necessary before endeavouring to design an appropriate code, since designing a program may entail the deployment of numerous mental representations in conjunction with a solid understanding of the computer language (He, Zhang & Li, 2021).

Routines are enclosed expressions comprised of procedures and functions that can be executed from anywhere in a code (Havenga & Mentz 2009). A routine that returns a value after execution is called a function. A routine that doesn't result in a result is called a procedure (Havenga & Mentz 2009). Because they return a value, function calls can be utilized as expressions in assignments and operations. For example, i := SomeFunction(x); calls SomeFunction and assigns the result to i. Function calls cannot appear on the left side of an assignment statement (Havenga & Mentz 2009). Procedure calls - For example, DoSomething; calls the DoSomething routine; the return value of DoSomething is discarded if it is a function. Procedures and functions can recursively call themselves (Havenga & Mentz 2009). Parameters are a type of variable that is declared in a subroutine and is used to pass a value into it. Arguments are the values that are passed in a function call (Havenga & Mentz 2009). In Delphi, we can declare functions/procedures using parameters such as the ones listed below (Havenga & Mentz 2009).

```
function Nothando(parameterList):
returnType; directives;
  localDeclarations;
```

```
begin
  statements
end;
```

where Nothando is any valid identifier, returnType is a type identifier, the statement is a sequence of statements that execute when the function is called, and (parameter list), directives, and localDeclarations; are all optional. The research attempts to unpack unfamiliar settings in functions and procedures and repack distinct settings by going from simpler meanings [concrete] to complex meanings [abstract] using the semantic wave construct of the Legitimation Code Theory. With an emphasis on functions and procedure programming activities, semantic wave theory is a simple yet effective way of teaching concepts in the context of learning (Maton, Hood, & Shay, 2016). What makes a good explanation or learning experience, whether textual, multimedia, or an online lecture, is of importance to the semantic wave (Maton, et al., 2016).

1.4. Research Aims

The main aim of this research is to unpack the mingling of functions and procedures settings and repack distinct settings by going from simpler meanings [concrete] to complex meanings [abstract] using the semantic wave construct of the Legitimation Code Theory. The legitimate code theory (LCT) through semantic wave will assist to lessen the misconception and come to an understanding of grade 10s functions and procedures in programming in the South African context.

1.5. Research objectives

The research sets out to:

Identify and discuss factors that compel learners in grade 10 to mingle the use of functions and procedures in Information Technology (IT) programming.

Examine the specific methods that IT teachers are using to teach functions and procedures in programming.

Evaluate the extent to which the methods used in teaching functions and procedures in programming contribute to grade 10 learners mingling the two concepts.

Propose the way the conception of the semantic wave of legitimate code theory can assist teachers to model the concept of functions and procedures in programming grade 10.

1.6. Research question

What makes learners in grade 10 mingle the use of functions and procedures in Information Technology (IT) programming?

1.6.1 Sub questions

1. What methods are IT teachers using in teaching functions and procedures in programming?
2. How are the methods used in teaching functions and procedures in programming make learners mingle the two concepts?
3. How can the Semantic wave of Legitimate Code Theory help teachers to model the concepts functions and procedures in programming?

1.7. Rationale of the study

This study is intended for information technology (IT) educators teaching programming in South African schools. During my student and teaching experience years it has become apparent that students find programming difficult especially the concepts of functions and procedures, since learners often engage with this concept right at the beginning of learning to program. I have decided to peruse research in this specific topic in hopes of a way to alleviate the challenge learners face as well as proposing a way to teachers that could lessen the misconceptions encountered learning functions and procedures and use better teaching methods that will improve learner understanding in programming. By utilizes the semantic waves of LCT to help learners understand functions and procedures. The concept of semantic waves breaks down technical language by moving from abstract meaning unpacking meaning by using everyday language as well as everyday examples then repacking meaning by relating the everyday language and examples to its original context. Refer to the diagram of semantic waves below (Diagram A). Grounds for the interrogations inspected as well as the conjectures reached ought to enlighten the gaps in the study.

1.8. Conclusion

As evident in this chapter, programming is a language just as any language it requires meaningful understanding, learners need to do more than simply memorizing concepts, they are also expected to understand and be able to execute those concepts in a practical context.

Failure to do so results in failure to implement running codes, which goes against the main point of coding or programming, meaning that the main objective of learning to program will not be meet. In addition to that, learner's failure to understand concepts in programming such as functions and procedures in Information technology implies that the goal of the National curriculum statements for IT is not meet by teachers or learner. Which remains a problem because this would result to no professional programmers produced in South Africa, which will negatively impact the economy since more people from external countries will be fulfilling programming occupations in a country that failed to produce more of their own.

Chapter two: Literature Review

2.1 Introduction

This section reviews what other authors have researched in programming, mostly at the high school level. Since programming is a very complex discipline, it possesses several challenges to learners understanding as well as difficulty on the educator's part to teach the subject in a level to which learners can comprehend. As a result, this chapter further highlights the misconceptions students face when writing code that involves functions and procedures in any programming language.

2.2 Programming language

Information Technology (IT) was introduced the South African education system as a discipline that was primarily centred on problem-solving and incorporating information and communication technology with public and commercial technologies to compete worldwide (Koorsse, Cilliers, & Calitz, 2015). This approach was made with the intention of strengthening continuous integration and engineering services by training upcoming professionals in the IT space with with the necessary capabilities (Havenga, & Mentz, 2009). The NSC outlines important teaching and learning practices and expected student achievement and establishes the basis for attaining the set of goals (Department for Education, 2014).

IT provides learners the opportunity to partake in tasks that are targeted at problem-solving. Problem-solving requires that learners become rational thinkers, collect, communicate, and manage detail also reinforces the invention of software packages employing the latest tools, as well as the growth of knowledge and comprehension of the interpersonal, institutional, along with other potential ramifications of computer use. " IT was formally introduced into South African schools for Grade 10 in 2006 and the first Grade 12 paper was written in November 2008" (Havenga & Mentz, 2009, p. 76). This further proves that IT was introduced in South African education at a later stage to other countries such as America where it was introduced in 1776. IT is among the recognized disciplines for tertiary admission provided learners meet the admission score the respective universities they applied for (Department of Education, 2003).

The south African school curriculum (NSC) specified the necessity for idea of mobility and intelligibility to be maintained (Department of Education, 2003). Which implies that for

learners who have completed matric need to meet the minimum score for the Information technology subject to be able to register in either colleges or universities in programming related courses. The objects of the curriculum for programming correlate favourably first entrance students at tertiary education level, hence the challenge for learners to excel in schools and universities (Department of Education, 2003).

Programming is the art of writing (coding) instructions to the computer, whereby the computer receives those instruction and carry out the commands as requested by the programmer (Robins, Rountree, & Rountree, 2003). The expectation is that decoding is what's required to attain expert-level skills in coding (Robins, et al., 2003). Its basic understanding that to engage in as well as practice any form of programming one needs to have at least some knowledge of how to operate a computer (Robins, et al., 2003). Program is either that from applied science or the other from the humanities such sector (Robins, et al., 2003).

Function and procedures are part of programming. Functions can be thought of as extraordinarily important in programming. It would be very hard to do computing programming without functions (Felleisen, Findler, Flatt & Krishnamurthi, 2004). The simplest way to think about functions is to think of Input, processing, and output or input, function, and output (Zelle, 2004). So essentially a function is something in which you put input and get some output (Robins, et al., 2003). On the other hand, a procedure is something that takes input and produces output as a result, a procedure allows us to use a small amount of code to do many different commands (Zelle, 2004. The use of functions and procedures confuses learners within the programming code, in most cases learners interchange the use of these concepts, thus this research intends to resolve this problem at an early stage of programming.

2.3 Using function and procedures in Delphi

Since south African schools that offer information technology focuses in either Delphi or Java programming languages below is a summary of how functions and procedures are used in either Delphi or java. Since, the calling of a function in a program fulfils the role of returning a value, a function also holds the potential of being used as phrases in assignments and operations. For instance, Y: =ThisFucntion (I) invoke ThisFunction as well as returns the output to Y (Felleisen, et al., 2004).

On the other hand, a procedure calls instead of returning a value. Both the calling of functions and procedures can qualify as complete statements if advanced syntax is implemented in the

program (Felleisen, et al., 2004). For instance: the function 'ThisFunction' the value that it returns is ignored because it is missing certain characters from the phrase that discards to fulfill the role of a function. However, DoSomething calls the DoSomething procedure recursive calls onto oneself are possible in procedures and functions.

2.4 Declaring functions and procedures

Declaring a function or procedures requires main elements to be indicated, such as the title, number or numbers, parameter accepted by either of the two concepts and in the case of a function the specific output it returns needs to be declared in beginning or header of the code or program (Franklin, Hill, Dwyer, Hansen, Iveland & Harlow, 2016). What follows the declaration is for the programmer to create a section of code that runs when either function or procedure is invoked and failure to follow or use the correct syntax results in failure to implement running code (Franklin, et al., 2016).

2.5 Procedure declaration

Procedure procedureName(parameterList); directives; localDeclarations; is the format for the declaration of a procedure in a program.

```
begin

 statements

end;
```

where *procedureName* is any valid identifier, *statements* is a sequence of statements that execute when the procedure is called, and *(parameterList)*, *directives;*, and *localDeclarations;* are optional.

Here is an example of a procedure declaration:

```
procedure NumString(N: Integer; var S: string);

 var
   V: Integer;
begin
V := Abs(N);
S := '';
```

```
repeat
S := Chr(V mod 10 + Ord('0')) + S;
V := V div 10;
 until V = 0;
  if N < 0 then S := '-' + S;
 end;
```

Given this declaration, you can call the NumString procedure like this:

```
NumString(17, MyString);
```

This procedure call assigns the value '17' to MyString (which must be a **string** variable).

You can utilize constants and other names declared in the localDeclarations section of a procedure within a statement block. The parameter list generates a group of instance variables, so avoid attempting to declare the parameter names in the local Declarations section.

2.6 Function Declarations

Like a procedure statement, a function declaration also provides a return statement and return value. The following format applies to function declarations: where function Name is any legal identifier, return Type is a type identifier, statements is a list of statements that are executed when the function is called, and (parameterList), directives; and localDeclarations; are optional.

The same guidelines that guide procedures are also applicable to the function's block of code. You may use any identifiers within which scope the implementation of functions occurs, as well as variables and other identifiers declared in the localDeclarations section of the function, parameter names from the parameter list, and other identifiers. Additionally, the predefined variable Result and the name of the function both serve as special variables that store the function's return value.

2.7 Misconceptions about Functions and Procedures

The mingling of functions and procedures is among the many misconceptions that make programming a challenge for learners to comprehend. Programming is not a new phenomenon, it is challenging on its own, this contributes to poor achievements by students as well as some students changing streams of subjects in schools and those at the tertiary

level either drop out their modules or drop out from the institutions (Cheah, 2020). The primary challenge that contributes to learners' impairments is the learner's inability to carry out problem-solving abilities when tasked with solving programming codes (Cheah, 2020).

To address major challenges relating to functions and procedures in programming, learners must have a knowledge of concurrent coding languages that goes further than syntax and execution. In the early phases of coding instruction, a deeper understanding of the context from the cognition abilities is required, according to previous studies (Ledgard, 1971). As far as the process of teaching programming is concerned, results from previous studies indicate that traditional books intended for teaching programming were found unsuccessful. This is due to the reasons that these resources comprise information on how a program is developed using difficult terminology that is hard to comprehend for the learner (Cheah, 2020).

Additionally, students' sentiments about studying computational thinking are impacted by their negative stigmatization of programming (Ledgard, 1971). As a result, this reduces the learners' inherent incentive to keep working hard to thrive and master advanced programming skills (Cheah, 2020). The reason students hold a negative perception of programming is due to information they had gathered from other students who have taken the programming courses in previous years. Students think programming will be challenging because they have a bad idea and the wrong perception (Cheah, 2020). As a result, its effects lessen intrinsic drive and do not encourage people to learn for success (Cheah, 2020).

Results from previous research have proved that the obstacles that hinder the mastery of programming are detected as early as the initial stages of programming (Cheah, 2020). The misconceptions that learners encounter when engaging with difficult concepts when coding (Cheah, 2020). These difficult concepts include algorithms, control functions, functions, and procedures. Another contributing factor to learners' mingling concepts in programming is the lack of teachers structuring activities in ways that develop intellectual faculties that promote high order thinking as opposed to basic forms of understanding (Cheah, 2020).

Programming requires some background information from other subjects, especially maths. Students who lack most of this knowledge are often left out, they get confused and

discouraged along the way (Cheah, 2020). Previous empirical research was able to provide evidence that the knowledge required to solve mathematical tasks was connected to the skills related to solving coding tasks in programming. This is further confirmed by the findings of Byrne and Lyons (2001), who concluded that learners who have trouble comprehending basic concepts in calculus, number theory, algebraic, and trigonometric principles have a hard time turning textual problems in programming into calculations.

Mannila, Peltomäki, and Salakoski's (2006) findings imply that a less complicated computer programming language leads to decreased grammatical and semantic blunders. The report's evaluative portion found no drawbacks to learning to program in a basic language before progressing to an even more intricate one (Mannila, Peltomäki & Salakoski, 2006). This shows that a simple language can be used to teach coding as a fundamental subject as well as to teach fundamental abilities to aspiring specialists.

The report referred to a category to evaluate challenges that learners encountered in learning complex topics in programming, these categories were learners' ability to implement codes, the relevance of their descriptions of codes, handling errors of the codes, and the way they used the programming software (Mannila, et al., 2006).

The findings of the study indicated that there were lexical and logical flaws. Programs written have significantly reduced syntax mistakes than Java programs did (Mannila, et al., 2006). The most frequent syntax mistakes in Java code were losing parentheses, skipping semicolons, using variables that weren't initialized, and misspelling words (Mannila, et al., 2006). In Python, the errors were caused by omitted colons or by single words. Other students lacked the skill to debug their codes for errors (Mannila, et al., 2006).

According to Zhang (2021)’s study students' programming difficulties continued when they advanced to higher education institutions. The findings indicated that Java programming is a complicated task for students in higher academic institutions since they never come close to mastering its competencies (Zhang, 2021). Without a basis, it is considerably more challenging for students to grasp the material because there are hundreds of application development principles (Joorabchi, English, & Mahdi, 2016). Also, several institutions and organizations are hampered by their timetables and refuse to make available certain software-related courses (Joorabchi, et al., 2016).

The study suggests that for teachers to minimize some of the challenges that students face when teaching functions and procedures, they should ensure that there is no confusion in their objectives, and their objectives should be straight to the point (Qian & Lehman, 2017). Teachers should use the learning result classification method to analyze the teaching target components when describing the goals (Qian & Lehman, 2017). When describing the goals, teachers should investigate the instructional focus sections using the learning result classification technique (Zhang, 2021). This will enable them to examine the appropriateness of the context, the focus of the topic, and how learners respond to it, this will also assist in formulating the appropriate criteria at which to structure instruction and learning (Alvarez & Scott, 2010).

Due to the complex nature of the subject or a lack of sufficient tools to guarantee individual attention from professors, many students have difficulty understanding (Santos, Tedesco, Borba & Brito 2020). The number of students compared to the number of teaching staff and the diversity of the participants provide challenges when it comes to cooperative learning, making it challenging to create programming procedures where everyone learns (Santos, et al., 2020). As a result, coding modules have frequently had significant drop-out rates. The learning style used is one of the key elements in raising the calibre of teaching programming, to address these problems teachers would have to decrease the difficulties frequently encountered and implement new pedagogical approaches (Alvarez & Scott, 2010).

2.8 Information technology in South Africa

There's a total of 29 school subjects taught in High schools according to the NSC (Koorse, Calitz & Cilliers, 2010). The Information technology subject enables learners' opportunities to gain skills and partake in tasks that targeted at problem-solving which calls for critical thinking, collaboration, and systems integration (Scheepers & de Villiers, 2000). Another major aspect that is emphasised in Information technology is the creation of computer applications employing concurrent technologies and software packages (Nkoala and Matsilele, 2021). IT in South African secondary schools has earned a reputation as a difficult subject because of the difficulty learners experience in understanding programming, stereotypical view in schools surrounding IT is that is a challenging subject, as a result grade 10 learners typically neglect choosing the subject and those that do choose it end up substituting the subject for one (Nkoala and Matsilele, 2021). Learners would consider less

challenging like subjects like CAT prior to proceeding to following grade 11 and 12, these impacts negatively on South Africa's ability to potentially produce high number of south African individuals IT related professions (Asongu, Orim, & Nting, 2019). This proceeds the growing tendency of waning enthusiasm towards programming-related careers in the past years (Asongu, Orim, & Nting, 2019).

Amongst the key knowledge and skills outcomes in the NSC for Information technology shareware and computer programming are the most prioritized (Koorsse, et al, 2010). It is claimed that the Information Technology (IT) subject taught in South African schools is a challenging one (Jenkins, 2002). The percentage of learners choosing to peruse IT beginning of High school years and those choosing to further continue with it throughout high school till higher education diminishes (Jenkins, 2002). The difficulty lays in the practical (programming) as opposed the theoretical aspect, since the practical aspect is the most prioritized it holds the most marks (Intisar & Watanobe, 2018). Therefore, learners who perform badly in this portion of the subject fail to attain desired results for the subject as expected by tertiary institutions (Intisar & Watanobe, 2018). The time stipulate to cover workload for IT is not enough for teachers to covers content sufficiently to guarantee that learners acquire the fundamental ideas needed to gain a through comprehension of concepts (Koorsse, et al, 2010). Also, the lack of teacher's growth development program by the department of education (Koorsse, et al, 2010).

2.9 Difficulty of programming

According to Koorsse, et al., (2010), grade 10 learner's responses to a questionnaire taken proved most of the learner's response attesting to the fact that procedures (or methods in Java) and functions to be a rather complex concepts in the Information technology subject. In addition to procedures and functions, parameters have historically proven challenging to master (Intisar & Watanobe, 2018).

"Procedures and functions require learners to change from coding in an event handler (such as a button click, or form create) to creating their own methods and calling these methods when needed" (Koorsse, et al., 2010, p.13). Pupils should comprehend code reuse and modularity to effectively grasp functions and processes. All those are ideas that object-oriented coding can expand upon (Intisar & Watanobe, 2018). It is necessary to conduct extensive research to determine whether the complexity in comprehending procedures and

functions is due to how these ideas are really implemented in computer languages or to an incomplete interpretation of these ideas' intended purposes (Intisar & Watanobe, 2018).

Respondents noted that troubleshooting their programs was tough, regardless of the coding knowledge or talents they possessed (Intisar & Watanobe, 2018). Evaluating if a learner would be able to fix any coding flaws depends on several significant aspects, including their coding skills and debugging skills (Intisar & Watanobe, 2018). Failure to debug code may discourage a learner, which will decrease their enthusiasm and passion for programming, Hence, learners require support or methods to identify problems in their programs (Alvarez & Scott, 2010). When the student employs methods and resources to recognize the inaccuracies on their own, that is one type of support, programmers can identify syntactic mistakes with the use of compiler error messages (Alvarez & Scott, 2010).

If individuals find it difficult to comprehend the programming language or use the correct variables for the correct results, they won't be able to properly fix several programming mistakes in the absence of additional help (Forišek, 2010). The systems engineering technologies employed in the academic institutions are intended for use by qualified IT professionals (Forišek, 2010). The routines and language error messages offered to help coders troubleshoot their code are frequently complex for learners who have recently began learning how to program (Forišek, 2010).

The teacher or other students are the second source of help that students seek out, when many pupils in a class struggle to figure out the solutions on their own, the teacher must devote the entire session going from student to student, pointing out mistakes and providing explanations (Tucker, 2010). Helping students in the classroom does give teachers a chance to determine what areas certain students are having difficulty with and to gauge their understanding of the concepts (Tucker, 2010). Teachers' ought to be cautious regarding helping students with trivial mistakes like omitting punctuation marks too early. Instructors really aren't encouraging students to build overall debugging skills if they do not have a method for helping students troubleshoot their own code (Tucker, 2010).

Additionally, the IT instructor may not always be able to help every student throughout the class instruction, which often lasts shorter than an hour (Armoni, Benaya, Ginat, & Zur, 2010). If they are having trouble identifying problems in the activities, many students tend to stop,

additionally, the instructor isn't always available to go over every resolution with the audience, therefore, it is up to the students to be certain they comprehend the answers and try to finish the tasks repeatedly by themselves (Armoni, Benaya, Ginat, & Zur, 2010). It is therefore highly challenging for IT teachers to make sure that students have enough exposure with coding tasks and workouts due to the limited time provided for the course and the enormous quantity of knowledge that must be taught (Armoni, Benaya, Ginat, & Zur, 2010).

2.10 Reasons for the decline in IT

There are clearly countless causes for the reduction in enrolments, and public perception is to blame for the greater part of them (Jacobs & Sewry, 2010). Popular myths about coding include the following: the subject is only for smart and exceptional people; programming in an innovative master level vocabulary is the sole viable career; and portioned alternatives eliminate the requirement for a massive number of software engineers (Jacobs & Sewry, 2010). Often these sceptics presume that drastic market research can motivate learners' desire to participate in IT (Jacobs & Sewry, 2010).

Half a century ago, constructing a desktop do anything was exhilarating; presently, it appears that vastly complicated software packages could be established with a plethora of component-based devices and no exhaustive coding skills whatsoever (Corney, Teague & Thomas, 2010). According to Corney, Teague & Thomas (2010), machine learning has forgotten its "allure," as well as the shortfall seen between academic and commercial industries is widening. Numerous different possible explanations for the drop in admissions can be identified with in academic abilities of educational institution along with the way they aid programming related causes (Corney, Teague & Thomas, 2010). One of the primary reasons pupils are getting bored is the failure of the system of teachers, scholars, and professionals to interact a cohesive and persuading discourse regarding technology management instead of merely future technologies (Meerbaum-Salant, Armoni, & Ben-Ari, 2010).

The worldwide downturn in programming course admissions suggests a significant dearth of learner motivation in programming related courses (Meerbaum-Salant, Armoni, & Ben-Ari, 2010). The loss of learner enthusiasm to peruse programming presupposes fewer student numbers, implying that academic industries are poorly managed in regard to their capacity to edify Information Technology (IT) professionals (Meerbaum-Salant, Armoni, & Ben-Ari, 2010).

The ever-changing spectrum and financial system act as a hindrance to the relatively stable school system, whilst also offshore outsourcing drives employment out of the country and away from skilled locals (Patton, 2020). Schools and other educational institutions are not keeping up with the rapidly changing economy, which requires an increasing number of highly trained programming occupations (Patton, 2020).

Reportedly, only 20percent of respondents of pupils will graduate with a bachelor's that will meet the requirements them for recruitment in the computing marketplace (Patton, 2020). South Africa is undergoing a disconcerting Computer science shortfall, to rising prices much further outstripping this same state's production of Information Technology trained professionals (Patton, 2020).

2.11 Concerns regarding information technology

Coding is considered as a rigorous cognitive endeavour that calls for a diverse variety of critical thinking capabilities, spite of significant study into the subject, the graduation percentage for undergraduate programs is still poor often at higher institutions (Clements, & Braman, 2021). According to Clements & Braman (2021) multiple organizations claim a fall in the percentages for beginner programming majors, at the end of their initial classes, most learners would not grasp what to do to code. Some pupils lack the information and abilities necessary for resolving issues, researchers find that most Information Technology students are only marginally capable of examining a small snippet of code in a comprehensive fashion, and that they do cannot understand the fundamentals of coding well enough to do simple tasks. Chiang (2018) point out that there hasn't been much comprehensive examination of the essential intellectual and educational complexity associated in instructing and learning how to program. There is absolutely no denying that learners battle to accomplish the goals of their coding, and this problem is present in secondary schools as well (Chiang, 2018).

Each country has a unique national curriculum, there should be more to computer programming (also known as information technology) than just coding information and expertise (Luxton-Reilly, 2016). In just about all circumstances, the emphasis is on educating pupils coding techniques that will assist them become proficient coders in a particular language (Luxton-Reilly, 2016). Throughout this sense, education institutions frequently let

the syntax of modern languages of programming to define their syllabus instead of fundamental design fundamentals and problem-solving (Luxton-Reilly, 2016).

Whenever the achievements in the South African Curriculum are analysed, identical outcomes listed in the UK curriculum can be found, nevertheless, a closer examination of the script reveals that 60% of the desired achievements, or coding, is where South Africa places the most attention (Benner, 2006). It is quite difficult to gain a comprehensive comprehension of the subject matter, moreover, the syntax of the scripting language in use is given a lot of importance (Benner, 2006). Therefore, creating a syllabus for high school students could be difficult, the conceptual level of secondary school students and the stage of intellectual growth required for learning technical knowledge must be explored (Benner, 2006).

Coding is regarded as a challenging cognitive endeavour that calls for a wide range of critical thinking abilities, despite extensive investigation, the pass rate for introductory courses is still unsatisfactory often at the university level (Xu & Rajlich, 2004). Multiple institutions record increased failure rate for new entrants programming students, just at end of their initial classes, most individuals would not be able to code (Xu & Rajlich, 2004).

Some pupils possess the information and abilities necessary for problem resolution, researchers find that the majority of computer science learners are only marginally capable of scrutinizing a small code fragment in a methodical fashion and that they do not understand the fundamentals of coding well enough to do simple tasks (Pardaboyevich, Abdunazirovich & Saydullayevich, 2020). Pardaboyevich, et al., (2020) bring out that there has not been much comprehensive examination of the underlying psychological and academic challenges involved in educating and learning of programming. There is no denying that pupils struggle to meet the objectives of their program, and this problem is present in secondary schools as well (Pardaboyevich, et al., 2020).

2.12 Conclusion

This chapter focused on teaching of Information technology (IT), particularly programming in South African schooling. It also provided some cues to learners on the opportunity they may partake in tasks that are targeted at problem solving, rational thinking, the collection and communication of data. The use of functions and procedures in Delphi as well as are

how they are declared differently in either Delphi or Java was discussed. Furthermore, it also focused on the misconceptions about functions and procedure. The significance of Information technology in South African schools, the difficulty of programming, the reasons for the decline of IT and concerns were discussed.

Chapter Three: Theoretical Framework

3.1 Introduction

This chapter contains the main theory that will guide this research. The Legitimate code theory (LCT) more especially the semantic waves notion of LCT. The semantic waves of legitimate conde theory contain How context dependence and complexity shape academic discourse which will serve as proposed pedagogical instrument to to unpack the mingling of functions and procedures settings and repack distinct settings by going from simpler meanings [concrete] to complex meanings [abstract]. The semantic wave and its principles are considered important for this research.

3.2 Theoretical framework

This study made use of the Legitimate code theory (LCT), but it employed the semantic wave notion of LCT that was created by (Maton, 2014) as a framework for the analysis of knowledge and instruction and is currently being applied to examine an expanding variety of various behaviours, It incorporates concepts from numerous disciplines, namely morphology, hermeneutics, science, and sociology, and it draws upon theories of Bernstein and Bourdieu (Harker & May, 1993).

In his study, Bernstein has repeatedly expressed the need to develop a connection among micro and analysis: The goal of the project is to develop a terminology that always allow for the integration of macro and micro levels of analysis as well as the retrieval of the macro from the micro in a dynamic world (Harker & May, 1993). This same initiative may be described as an ongoing effort to gain an understanding of the fundamentals of information exchange that govern the genuine formation, dispersion, propagation, and alteration of awareness as well as the legitimate reproduction of dominant cultural subgroups and a given state apparatus (Harker & May, 1993).

An endeavour has been made to comprehend the nature of representational regulation in brief Bernstein has used the conception of the notions of categorisation and structuring, accessible and hidden teachings, to investigate in an intellectual method how representational control is applied over the learning to accomplish this goal (Bernstein, 1990). A code is a normative paradigm that underpins numerous apparatuses, including education and pedagogy Bernstein's co-articulated premise (Gorder, 1980).

3.3 The Legitimate Code Theory (LCT)

The legitimate code theory can disclose what it takes to succeed, or what we refer to as "game rules," and it can be communicated to learners, it also strives to empower teachers by providing a set of tools for strengthening teachers' thinking, as teachers are the ones who deal with students in the classroom (Maton, 2014). It also allows us to examine identification and demonstrate which teaching methods are most effective for imparting various types of knowledge to various types of students (Maton, 2014). Dlamini & Dewa (2022) view the Legitimation Code Theory's (LCT) Semantic Code as offering unique insight into the ways in which programming skills may be used to produce a working and meaning codes as well as how to make teaching programming languages enjoyable for learners. LCT enables the diagrammatic representation of systematized ideas, improving communication (Dlamini & Dewa, 2022). To uncover epistemic and semantic principles that might increase students' pleasure of programming languages, LCT may offer a conceptual framework for doing so (Dlamini & Dewa, 2022). LCT is a framework for examining what determines and confirms a positive learning experience, to put it another way (Maton, 2013).

The explanatory framework the study employs is the concept of semantics, which is built on the basis that everything we do, including experience and understanding, claims, and knowledge practices are based on the principle that everything we do is (Maton, 2014). There is an element of generality here, as well as some level of technology. As a result, the concepts of Semantic Gravity (SG) and semantic density (SD) are used to investigate this theory (Maton, 2014). And these ideas enable us to ponder what forms knowledge takes, as well as how these forms of knowledge integrate, alter, change, and move through text and time (Blackie, 2014). They are highly abstract principles that must be executed in a specific situation, and they must constantly be operationalized (Blackie, 2014).

Semantic Gravity (SG) is known as the level to which meaning connects to its context, it can be either strong or weak along strengths (Maton, 2014). As a result, we would contend that stronger Semantic Gravity has an independent context while lesser Semantic Gravity has a more abstract context (Maton, 2014). By extrapolating generalizations from a particular instance, context, or case, Semantic Gravity can be reduced. Additionally, we can discuss how Semantic Gravity may vary or move dynamically over time (Maton, 2014). Additionally, you might be able to enhance the semantics (Maton, 2014).

Semantic density (SD) is a term used to describe the strength or weakness of meaning condensation (Maton, 2014). Therefore, an image or meaning, a category of item, can be more potent when more meaning is jammed into it than when the density is higher and there are fewer meanings condensed into it (Maton, 2014). This can also be dynamics so we can talk about movement over time, when strengthening SD we condense and put more meaning to the subject, object, or technical term (Maton, 2014). The weakening of semantics density refers to unpacking meaning (Maton, 2014).

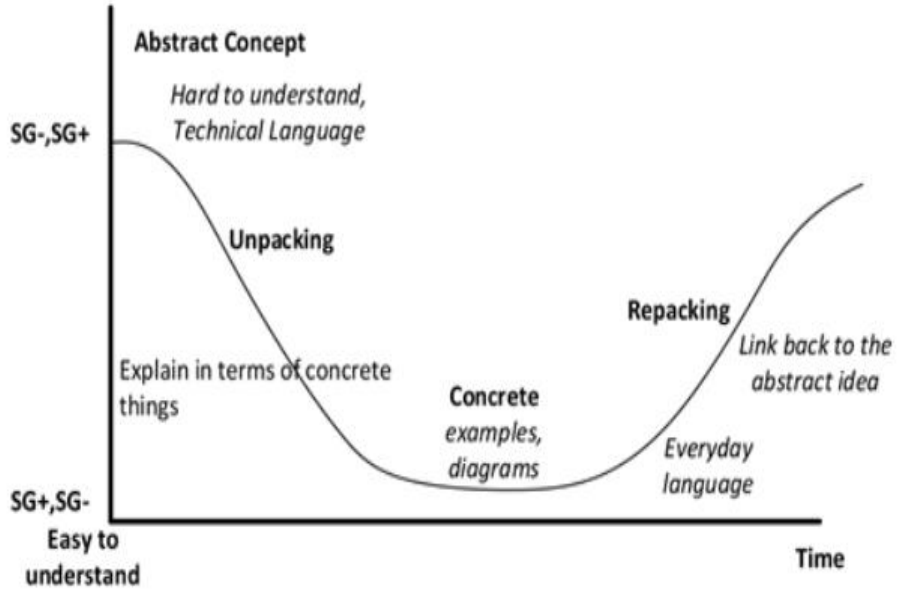
Then we talk about putting them together, and how in semantics we have more concrete and simpler meanings which is the use of everyday language and real-world examples (Maton, 2014). The Semantic Gravity can be weakened this happens when beginning to discuss the idea of more obstruct but they are still quite simple in their meaning (Maton, 2014). This is often the notion of generalization, the density is strengthened upon getting more complex meaning with obstructing ideas and getting the technicality there is and the abstract concept happening, but also, we can strengthen the gravity where we start talking about concrete examples of complex meaning (Maton, 2014). It also helps to speak of the notion of specialized language with real-world examples (Maton, 2014).

Since the concepts of functions and procedures are abstract and academic language, such abstract language would be broken down to everyday language accompanied by real-world examples without losing its academic sense to help learners easily comprehend the two concepts. By so doing this, we would be weakening the Semantic Gravity.

The reason this is useful is that students do need more than one code, they need to be able to unpack and repack them (Maton, 2014). They need to be able to move across these ideas, often in readings and their writing they need to be able to trace the movement recognized to help them understand and build their cumulative understanding and the ability to work with abstract terms (Maton, 2014). So that is a tangible example of why semantic codes could be useful when working with text and vocabulary we can use this concept to talk about changes over time which is where we get the notion of semantic waves (Maton, 2014). So, we can talk about meaning that is often hard to understand often abstract they're often technical for example to understand that we often need to unpack it, we need to think about concrete examples of these so we might be decreasing the density

sort of unpacking the meanings in there or we can give a real-world example which is and an example of high gravity (Maton, 2014).

What is less desirable for teaching and learning is what we call a high semantic flat line, so this is often the language of experts, it has a place in education regarded as expert communication however there is a reason for such technical language it is all about context but it often less desirable for what we need to be doing with students (Maton, 2014). Alternatively, we often have a low semantic flat line here the means often get locked into a particular situation we presented in language so that students can easily understand but they may not be able to abstract the meaning out of a particular context and transfer them to other contacts now coming we can also talk about. Below is a diagram illustrating the semantic weaves (Maton, 2014).



Semantic wave: Source: Waite, Maton, Curzon, & Tuttiett (2019).

Rusznyak (2020), conducted a study using Legitimate Code Theory, this study used semantic codes to analyze semantic differences in pre-service teachers' reflections using the original revised reflection prompts. The study discovered that pre-service teachers' reflections on accounts were Semantic Gravity, they found an equal amount of Semantic Gravity and semantic density Rusznyak (2020). This is relevant to the proposed study because in the study they used similar concepts that I am also going to implement in my study, those

concepts include the Legitimate Code Theory as well as the notion of semantic codes also referred to as semantic waves, Semantic Gravity, and semantic density.

3.4 Semantic Waves of Legitimate Code Theory

The first concept, Semantic Gravity (SG), describes how easily meaning can be lifted out of the context in which it was created, the more context-dependent the meaning is, the stronger its Semantic Gravity (Rusznyak, 2020). In the context of this study, stronger levels of Semantic Gravity would characterize reflections that focus on the details of a particular lesson, a weakening of Semantic Gravity manifests when entries draw on or generate generalized patterns of teaching that can be transferred across contexts or time (Rusznyak, 2020). In this study, Semantic Gravity allows analysis of the extent to which students confine their reflections to the visible events in a particular lesson, consider intangible aspects of the lesson, and whether they bring patterns or principles to bear on the lesson at hand (Rusznyak, 2020). When undertaking an LCT analysis, it is necessary to use a translation device that enables articulation between its analytic concepts and a coding system for how these concepts manifest in the empirical data of this study (Maton and Chen 2016). In this article, four levels of Semantic Gravity are defined, ranging from stronger (SG++) to weaker (SG--) (Rusznyak, 2020).

The second concept, semantic density (SD), indicates the degree of complexity expressed within the reflection (Rusznyak, 2020). A straightforward description of a classroom event would have a simpler meaning, indicated by a weaker semantic density (Rusznyak, 2020). When pre-service teachers interpret and explain classroom events, they add meaning to a description, strengthening semantic density. In addition, when they draw on formal concepts to explain classroom events, the reflection becomes connected to a wider body of theoretical knowledge (Rusznyak, 2020). Their reflection becomes more complex, strengthening its semantic density (Rusznyak, 2020).

3.5 Research Methodology

In everyday speech, the term "research" means the pursuit of information. Another way to think about research is as a methodical and rigorous search for relevant data regarding a certain subject (Snyder, 2019). Research is a form of artistic rational inquiry. Studying is an art (Bernard, 2006). It is a meticulous examination or study particularly when looking for new facts in any discipline of expertise, research is very much an interesting addition to the

amount of evidence that benefits towards its expansion. It refers to the search for reality via research, experience, analysis, and experimentation (Snyder, 2019).

In a nutshell, research is the process of seeking enlightenment through an organized, methodical approach to a subject, research also includes a methodical approach to conceptual creation and refinement (Gupta & Gupta, 2022). As a result, the term "research" pertains to the methodical process that includes stating the issue, developing a supposition, gathering the facts or data, analysing the evidence, and coming to certain findings that may take the form of remedies to the relevant problem or generalizations for a positivist approach (Gupta & Gupta, 2022). It is crucial to understand that a wide range of techniques can be used to conduct observation and experimentation and analysis (Kothari, 2004).

Nonetheless, it is both feasible and beneficial to research every relevant occurrence in a lab, every article's planning starts with choosing a problem and a technique of inquiry (Kothari, 2004). Such early choices are based on presumptions about the society, how research ought to be done, and what significant concerns, answers, and "evidence" standards are (Baker, 2000). Various research methodologies include both theory and procedure, there seem to be two broadly agreed methods: quantitative research and qualitative research (Baker, 2000).

Knowing and comprehending the methodological strategy that underpins whatever study is crucial since the selection of a study design affects problems that arise within the study, the methodologies selected, the quantitative assessments conducted, the implications drawn, and the aim of the study (Baker, 2000). Whether a study is quantitative or qualitative will affect the questions posed and the arguments made when examining different research (Walker, 1997).

Quantitative research is an investigation into a known issue that is built around the verification of a hypothesis, quantified using numerical data, and statistically analysed (Vishnevsky & Beanlands, 2004). Quantitative approaches seek to ascertain the validity of a philosophy's prediction implications, quantitative approaches are based on the premise that existence is neutral and independent of the investigator, allowing it to be examined scientifically (Vishnevsky & Beanlands, 2004).

The purpose of the value-free research is to create abstractions that add towards the hypothesis and assist the researchers foresee, interpret, and analyse certain occurrence, the

study relies mostly on rational modes of reasoning and concepts and ideas are evaluated in an aggravate order (Vishnevsky & Beanlands, 2004). The objective of a qualitative investigation process is to comprehend a sociological or psychological phenomenon from various angles (Patton, 2005). A nuanced and comprehensive image of the phenomena under study is developed through the process of qualitative analysis, that is also carried out in a non - contrived environment (Patton, 2005).

3.6 Assumptions Underlying Qualitative Methods

Numerous different realities—the author's, the experiences of the people being studied, and indeed the peruse or viewers reading and understanding the findings—exist in any scenario (Patton, 2005). Such viewpoints, or conversations, of respondents (i.e., participants), are incorporated into the analysis; the researcher engages with many of those he analyses and vigorously attempts to minimize their separation from him; and the analyst consciously identifies and concedes the value-laden nature of the research (Pathak, Jena, & Kalra, 2013).

There are several different sorts of research methodologies: mixed, qualitative, and quantitative (Khaldi, 2017). Khaldi (2017) asserts that whereas qualitative research is often affiliated to the humanities, quantitative studies are frequently linked to empirical evidence. Mixed-method research incorporates the strengths of qualitative and quantitative which is frequently used in experimental studies, especially in teaching and learning (Khaldi, 2017).

However, this study uses a qualitative research methodology due to reasons that the number of participants will be less than 10, it will comprise 5 teachers from schools that offer IT in the Johannesburg east district with a minimum of two years teaching IT and above, contact engagement is required with these participants (Aspers & Corte, 2019).

According to Khaldi (2017), qualitative research is a method for studying and comprehending the meaning that individuals and organizations attach to a sociological or psychological situation, emerging questions and techniques are part of the research process, typically, data is collected in the participant's environment.

Aspers and Corte (2019) argue that the goal of qualitative research is to answer questions about how to better comprehend the meaning and experiential components of people's lives and society. According to Aspers and Corte (2019), qualitative research is multimethod in nature and takes an expressive, phenomenological approach to its topic. This implies

qualitative researchers look at things in their natural surroundings, aiming to understand or interpret events through the lenses of the meaning's individuals assign to them (Aspers and Corte, 2019).

3.7 Research Paradigm

Many academics have varying interpretations of what is meant by the word paradigm. According to MacNaughton, Rolfe, and Siraj-Blatchford (2001), a research paradigm is made up of three components: a procedure, standards for authenticity, and a conviction about the nature of facts. The framework is referred to as epistemology, ontology, or even research technique by Neuman (2000) and Cresswell (2003). Conceptual paradigms are categorized by Mackenzie & Knipe (2006) as positivist (post-positivist), rationalist, hermeneutic, radical, emancipatory, analytical, practical and deconstructivism, postpositivist or hermeneutic, depending on how they are understood. In the postpositivist paradigm, cause and consequence govern the philosophy (Creswell, 2003).

3.8 Research Paradigms

The foundation of our perceptions about existence is referred to as ontology Scholars make assertions regarding existence, how everything functions, and what could be learnt from it, occasionally inadvertently (Khaldi, 2017). The metaphysical problem, A single, provable existence and veracity or historically created numerous possibilities, is just what prompts a scholar to ask what form of existence is genuine (Khaldi, 2017).

The field of metaphysics known as epistemology is concerned with the nature of information and how it is gained and evaluated the structure and manifestations of information, ways in which it is produced, and the way it is transferred toward other individuals are of significance (Khaldi, 2017). A researcher will talk about the potential and appropriateness of impartiality, variability, correlation, soundness, and comprehensiveness in response to the epistemic inquiry, following an explicit or implicit metaphysical ideology directs one to epistemological assertions (Khaldi, 2017).

A defined, philosophical process to the generation of data is what methodology is It alludes to the research and evaluation of data generation methods, the goal, planning process, procedure, or design influences one's decision regarding the study methodology (Kivunja & Kuyini, 2017).

Methods are certain ways to gather and interpret information, the design of the research topic and the author's theoretical perspective would determine the methodologies to be used (Kivunja & Kuyini, 2017). To be clear, using procedures does not require making philosophical or cognitive commitments (Kivunja & Kuyini, 2017).

The term axiology refers to the ethical problems that must be considered while developing a scientific project. It considers the conceptual model for arriving at worthwhile or moral conclusions. In Finnis (1980). Kivunja, C., and A. B. Kuyini (2017). It entails establishing, assessing, and articulating ideas of appropriate and unethical conduct in connection with the study. It considers the importance we will assign to the various components of the study, the respondents, the information, and the audience to which we will present the findings of our study (Kivunja & Kuyini, 2017).

There are three methods for conducting research study: Positivism, Interpretivism, Critical theory (Cohen, Manion, & Morrison, 2002). Positivism holds that true knowledge apart from people, It is unmediated by our perceptions and subject to unchangeable laws, realisticism is the metaphysical stance taken by positivists (Krauss, 2005). Positivists want to comprehend society in the same way they do the physical universe, occurrences have a causation correlation that, when proven, enables future prediction to be made with accuracy (Krauss, 2005). The sociology holds true for empiricism in the same way, since, actuality is context-free, investigators from many eras and locations will arrive at the same findings regarding a certain occurrence (Krauss, 2005).

Positivism is the objectivist's epistemic stance, researchers examine events that act separately of each other in an unbiased fashion; researchers need not tamper with or alter the occurrences that are being examined (Krauss, 2005).

A reaction to the overwhelming influence of optimism, interpretivism, the assumption that there is an unified, observable actuality that is separate from our experiences is condemned by interpretivism (Hua, 2015). This opposes hypothesis testing to use interpretive ontology, It rejects embracing any enduring, unchanging (or fundamental) norms through which reality may be revealed to all (Hua, 2015). Interpretivists, on the other hand, think that various truths are culturally produced, actuality and authenticity are made, not found, given that truth is

indeed filtered by our perceptions, it is impossible to comprehend truth in its entirety. Epistemology that is interpretive is arbitrary (Hua, 2015).

Observers never perceive objective world directly without it being tainted by their standpoints, attitudes, identities etc (Bunniss & Kelly, 2010). Vision is understood as an active constructive process of production rather than a silent process of portrayal, people engage in social interactions with one another and alongside community at large, giving many social phenomena labels and implications (Bunniss & Kelly, 2010). Researchers are not "separated" from the issue they are studying because they are an integral component of the social truth being studied (Bunniss & Kelly, 2010).

When there are several well-supported explanations for a specific hypothesis, currently no single explanation is handpicked as the "completely accurate" explanation or even as preferable to others (Rahi, 2017). Instead, the presence of countless explanations is acknowledged in addition to the fact that different authors have conflicting views on a particular problem (Rahi, 2017). Instead of seeking out facts and information that transcends circumstance and significance, interpretative research aims to comprehend how people see the complex reality they engage with (Rahi, 2017). An inescapable result of interpretive ontology is this idea of understanding. If one thinks in many socially defined truths, it implies that various individuals view these complexities from various perspectives (Rahi, 2017).

According to interpretive technique, human behaviour must be seen through the individuals' perspective as opposed to the researcher, interpreting human behaviour in the setting of their surroundings is the aim of interpretative analysis (Rahi, 2017). As with ethnography and case studies, interpretivists primarily gather empirical information about people through an extensive period (Holton, 2009). The method for analysing the information that is thus produced is inductive, meaning that the investigator looks for trends in the information that can be grouped into broad themes to comprehend a reality and form hypotheses (Holton, 2009).

Since people tend to perceive theory as emerging from information gathering and not as the driving force of research, interpretivists utilize the inductive technique rather than the mixed method (Holton, 2009). Data is typically sound, or film documented and primarily

conversational rather than quantitative to retain the happenings in a reasonably appropriate fashion for eventual data processing (Holton, 2009).

While descriptive statistics may be included, interpretive researchers use approaches that produce qualitative information instead of relying upon that (Walsham, 2006). Non-restrictive interview process with differing extents of formation (formalized open-ended interviews, moderately open-ended interviews, and group discussion interviews), findings, recorded notes, scribbled correspondence, transcripts, etc are instances of methods for gathering data that produce qualitative data (Walsham, 2006). Another set of standards to evaluate the trustworthiness of content analysis have been proposed, a study possesses trustworthiness, replicability, durability, and impartiality, it is deemed to be of high quality (Walsham, 2006).

Study aspects of the Guba and Lincoln performance metrics if the investigators are genuine and dedicated in continuous pursuit of an interpretation of the facts, the findings have relevance for individuals in various circumstances, and the stages and techniques of the study are thoroughly disclosed (Taylor, 2005).

It is accepted that there is a world, but it has been created by social systems that are influenced by cultural, governmental, economic, sexual, and religious aspects (Kincheloe & McLaren, 2011). Critical theory is biased from an epistemic standpoint since it operates under the premise that no issue could be studied objectively without influencing the observer (Kincheloe & McLaren, 2011). Prior to initiating an analysis, academic - related researchers work to be aware of their own epistemological principles and express them explicitly so that no one is mistaken regarding the epistemology and ideological weight they carry to the testing facility (Kincheloe & McLaren, 2011).

Research Design

This study employs a qualitative case study research design, which is a research method that allows a complicated occurrence to be investigated by identifying multiple components that interact with one another (Creswell & Creswell, 2017). The situation depicted is a real one, because of the research, individuals' experiences can be imagined (Creswell & Creswell, 2017). A researcher's research design is the framework for the methods and approaches he or she will use (Creswell & Creswell, 2017). Researchers can focus on research methodologies

that are appropriate for the topic matter and set up their studies for efficiency thanks to the design (Creswell & Creswell, 2017).

The qualitative case study research design is a research method that allows for the investigation of a phenomenon in its context using a variety of data sources (Baxter & Jack, 2008). This guarantees that the topic is not examined through a single lens, but rather through a range of lenses, allowing for the discovery and understanding of numerous aspects of the phenomenon (Baxter & Jack, 2008). According to Baxter & Jack (2008), a case study design should be considered when: (a) the goal of the study is to solve "how" and "why" questions; (b) you can't control the behaviour of those who are part of the study; (c) you want to fulfil those needs because you believe they are relevant to the phenomenon under study, or (d) the limits between the occurrence and the context aren't clear. A descriptive case study is a concentrated and thorough investigation in which hypotheses and concerns regarding a phenomenon are carefully examined and articulated from the start (Baxter & Jack, 2008).

Generally said, a research design is indeed the foundation or plan for an investigation that functions as a direction for gathering and analysing data. It is a process that is pursued to accomplish a thesis. A research design is a comprehensive strategy that outlines the techniques and steps to be taken to gather and analysed the necessary data.

The following objectives are achieved through the utilization of research designs: to reduce spending. As a result, it offers a strong foundation for the entire study. By creating an advance strategy for the entire research project, this maximizes the quantity of data provided while consuming the least amount of time, expense, and energy.

It enables the seamless scaling of the numerous scientific procedures, hence maximizing information production with the lowest amount of effort, energy, and expense. Gather pertinent information and apply appropriate techniques: Research design refers to the early selection of the approaches to be taken for gathering pertinent information and the approaches to be applied for their assessment, while considering the objective of the study and the affordability of workplace time and financial resources. The entire endeavour was derailed by inadequate research design planning. For give ideas a template: Since it enables the efficient execution of multiple research processes, research design is necessary.

It is like a template that we require ahead advance to plan the approaches to take for gathering the pertinent data and the procedures to employ for its examination to prepare a

research study. Like how a property needs a plan as well as a guide for more aesthetic and cost-effective building, research requires a strategy or a layout for it to run smoothly. To give a general insight towards other professionals an insight of the entire study process is provided by a research design, and we are able enlist the assistance and opinions of specialists thanks to the design.

3.9 Three general types of qualitative methods:

Although this study used a qualitative case study, the five qualitative methodologies are narrative approach, phenomenology, grounded theory, and case study. In a case study, the researcher investigates a particular image or occurrence (referred to as "the case") that is restricted in space and time (such as a curriculum, occasion, organization, or demographic relations) and accumulates in-depth information from a wide range of sources. procedures for collecting over a prolonged time. A case study is a narrative account of a person's knowledge and/or activities that is kept by a third party.

3.10 Data Collection

According to Morgan & Harmon (2001), is the methodical process of gathering and quantifying data on variables of interest to respond to research problem, make predictions, and assess results. Interviews will be my data collection tool (Morgan & Harmon, 2001). An interview is a series of questions that are asked orally by the interrogator and are typically answered by the respondent (Morgan & Harmon, 2001). A semi-structured interview was selected; this specific type of interview was chosen due to the reason that this it involves a mixture of structured and unstructured interviews. Hence the preference to probe further as opposed to strictly following a fixed set style of questioning, (Morgan & Harmon, 2001).

3.11 Population

The term "research population" also refers to a delineated group of people or things that are recognized to share common traits. Typically, every person or thing within a population shares some sort of unifying quality (Aliard, 1970). In simple terms, A population is an entire group that you want to conclude (Draugalis & Plaza, 2009). The Population of my study will be teachers in Schools that offer information technology from grade 10 to 12, the information technology is taught by teachers who have a minimum teaching experience of three years and more teaching information technology either in Delphi or java language (Draugalis & Plaza, 2009).

3.12 Sampling

This research uses the purposive sampling technique, where I will sample 5 out of 100 high schools in Johannesburg east district, that offer information technology and 5 teachers with a minimum of 2 years and above experience in teaching information technology A sample is a portion of the population that has been chosen to be representative of the whole, we need to take a sample because we can't research the full population. The use of a purposive approach is justified by the notion that, considering the report's aims and objectives, people who are experts, meaning that they qualify as teachers qualified to teach Information Technology may have special and valuable positions on the topics linked to the research question and, as a result, need to be part of the research sample or sampling process.

In addition to purposive sampling the research will use convenience sampling

'Probability' and 'non-probability' samples are two types of sampling methodologies (Acharya, Prakash, Saxena & Nigam, 2013). Probability sampling enables the researchers to extrapolate the sample's findings to the entire population. Simple random sampling, systematic random sampling, stratified random sampling, cluster sampling, and other types of probability sampling are all examples of probability sampling (Acharya, et al., 2013). A sampling frame is critical in probability sampling because random sampling from that frame cannot address the research problem if the sampling frame is not picked suitably from the population of interest (Acharya, et al., 2013). Only the actual population defined by the sample frame can be used to generalize. Convenience/purposive sampling, quota sampling, Snowball sampling, and other non-probability sampling methods are all examples of non-probability sampling (Acharya, et al., 2013).

3.13 Data analysis

The study used a qualitative data analysis which involves the collection and analysis of qualitative data to understand concepts opinions and experiences, this will be the analysis of the data obtained through focus groups (Information technology teachers from grade 10) (Morgan & Harmon, 2001). The discussions carry it out through semi-structured interviews available through the recordings I would have taken (Morgan & Harmon, 2001). I will then analyze this data and allocated it accordingly to my research questions since most of the data would be responses to my research questions (Morgan & Harmon, 2001). This study followed

the Interpretive Phenomenological Analysis (IPA) qualitative method of analysis. The main purpose of this analysis is to explore in detail how participants are making sense of their personal and social world to dig out an individual's perception or account of an object or phenomena. The research questions answered through IPA are usually framed broadly and openly there is no attempt to test a predetermined hypothesis of the researcher rather the aim is to explore flexibly and in detail an area of concern (Shinebourne, 2011). Using IPA, I examined how teachers approach teaching functions and procedures, drawing on the questions I pose to them about the different teaching methodologies they employ. Using their answers to those questions as a starting point, I will be able to determine how the teaching techniques lend a hand to students in grade 10 mingling the concepts of functions and procedures. Due to the need for a careful study of each transcript, IPA studies are typically conducted on small samples through purposeful sampling (Shinebourne, 2011). For this reason, I chose 5 IT teachers with a minimum tenure of three years teaching IT for my unit analysis. This is so that they can teach more effectively than new instructors and teachers in their first year of employment who have less than three years of experience.

3.14 ETHICAL CONSIDERATION

This study employed an empirical research methodology, this alludes to the fact that the study will involve human participants in the form of Information Technology Teachers in grade 10 with a minimum of 2 years and above of teaching experience. To protect participants information, application of ethical clearance from the university was made to get permission to engage and ethically conduct the research In addition, there is a permission letter from the Gauteng department of education (GDE) granting the researcher to access gather data from selected schools. To protect the participant's identity, no school names will be mentioned, neither the participants. Schools or participants names were pseudo-coded to hide the identity. The participation sheet provided the participants information that the interview is voluntarily so that they are aware that they are not being forced to participate. Also, if any of the participants wish to be excused from the interview process, they may do so at any point in the data collection process.

3.15 Conclusion

This chapter provided a detailed description of the research design as well as various methods employed by the study, using qualitative research methods. The primary data was collected through semi-structured interviews.

Chapter four data presentation, analysis, and Discussion

4.1 Introduction

This chapter presents data obtained from the participants. The participant's information is anonymized to hide their identity and their places of work. For example, teacher A, B, C, and D were coded as T1, T2, T3, T3 and T4. Then the school that teaches comes from were coded or named s1, s2, s3 and s4. (See Table 1)

Table 1

Teacher Name	School Name
T1	S1
T2	S2
T3	S3
T4	S4

The study attempts to (Identify and discuss factors that compel learners in grade 10 to mingle the use of functions and procedures in Information Technology (IT) programming, Examine the specific methods that IT teachers are using to teach functions and procedures in programming, Evaluate the extent to which the methods used in teaching functions and procedures in programming contribute to grade 10 learners mingling the two concepts and propose the manner in which the conception of the semantic wave of Legitimate Code Theory can assist teachers to model the concept of functions and procedures in programming grade 10.). The major categories that emanated from the data collection include challenges educators face, teaching methods, the distinction between function and procedures, reasons for learners mingling functions and procedures and learners are starters in programming and lacks some abstract mathematical logical thinking. The chapter highlights what the participants mentioned to help answer the research question (What makes learners in grade 10 mingle with the use of functions and procedures in Information Technology (IT) programming? What methods are IT teachers using in teaching functions and procedures in programming? How are the methods used in teaching functions and procedures in programming make learners mingle the two concepts? How can the Semantic wave of Legitimate Code Theory help teachers to model the concepts functions and procedures in

programming?) and achieve the aim of the study (to unpack the mingling of functions and procedures settings and repack distinct settings by going from simpler meanings [concrete] to complex meanings [abstract] using the semantic wave construct of the Legitimation Code Theory. The Legitimate Code Theory (LCT) through semantic wave will assist to lessen the misconception and come to an understanding of grade 10s functions and procedures in programming in the South African context).

4.2 Challenges educators face

Information Technology teachers for Grade 10s have a mountain-like challenges in teaching Grade 10 programming, as they are debutant in the subject. They need to work hard to recognise what difficulties learners face and to support them by developing learning materials for programming in different topics. Programming is not an easy subject to be studied. It requires correct understanding of abstract concepts. Many students have learning problems due to the nature of the subject. In addition, there are often not enough of resources and students suffer from a lack of personal instruction. This was reflected during data collection, and this is what T1 mentioned,

> *These learners are new into this subject, and they are confused. The subject on its own requires abstract logical thinking and most of our learners are coming from poor background with limited digital exposure.*

T1 noted that learners are new in the subject and the subject requires high order thinking. In line with the subject being new, the learners are not all possessing the digital technology tools such as laptops and smart phones. T3 echoed the similar sentiment, but however he added,

> *The classes are large and heterogenous and thus it is difficult to design the instruction so that it would be beneficial for everyone. In addition, in our school, learners share the machines that are there. They only do programming at school, and they cannot practice it at home as they do not have the machines with the programming software. This often leads to high drop-out rate in the subject.*

There is evidence from T3's comments, that the learners share the few desktop computers that are in the lab. In addition, this limits the learners to continue studying or practising computer programming outside the school walls. However, T2 mentioned that he has just graduated from university three years ago in computer science. He got the job for IT in schools. The challenge was that he was new to Delphi programming language as it is not offered at university level. This is his remarks,

I started learning Delphi at the same time teaching learners. This was a challenge as I would teach myself first before teaching learners. Where I did PGCE qualification, I was taught the methodology or educational theories only, not the content. In essence I was not confidence with the subject, and I feel I was not teaching them very well.

The teacher started knowing Delphi programming while on job. He did self-training while at the same time teaching. This on its own suggests that T3, was almost at the same level with the learners when started teaching Delphi programming language.

On a different note, T4 mentioned that the most challenge she faces is that learners lack mathematical logical thinking in programming aspects. He remarked,

These learners are required to demonstrate competencies in high-level abstract principles of programming and logic, such as program design and OOP principles, which are conceptually difficult. During the programming task learners to solve some mathematical problems and it is in this aspect that they display low knowledge in mathematics. In addition, they are still struggling to understand simple syntax rules. This problem tends to be exacerbated by the trend of learners to study independently, outside the classroom or in online modes.

4.3 The most challenging topics teaching IT to grade 10's

Despite advancements in methodologies and technologies, learning to code continues to pose lot of difficulties, therefore failure rates are indeed significant (Medeiros, Ramalho, & Falco, 2018). Scripting languages are intricate man-made structures. Like the syntactic principles underlying native tongue, scripting languages are comprised of a relatively limited number of components that may be assembled in a limitless array of ways. Students, however, seem to be at a loss over which component must be used in the programming syntax. They are composed of a relatively small number of components that can be joined in an infinite number of ways, much like the grammatical rules of natural language (Robins, 2019). However, learners seem to be in doldrum when it some to which element should include in the programming syntax. Functions and procedures prove to be amongst the most topics in information technology as evident by the IT teacher's response below.

T1 confirms that functions and procedures are among the most challenging topics to grade 10 learners. He stated:

So, I would say it would be procedures and functions, basically because when you start teaching grade 10s you start teaching them how to for examples use mathematical

calculations by using for example multiplication, division and addition. Then you get mathematical functions like pi for example that is about in-bult function and then you need to teach them how to use that instead of for example 3,14 in your code. Things like using the inverse procedure or the decrease was really challenging for them because now you no longer have like this static program where you hard code things you now have functions and procedures that you can use in order to eliminate those things. I think that was the challenge and that is why the kids found it challenging as well.

Most novice learners find programming difficult to comprehend. Such situation can also be ascribed to a variety of elements, including the subject's intrinsic limitations and learners' absence of previous expertise on how to solve math equations. Most of the problems learners face are caused by this crucial last element (Gomes, Carmo, Bigotte, & Mendes, 2006). T1 listed several concepts where learners face challenges. Operators and functions used in mathematics are included. The procedural statement, such as increasing or decreasing the count, is troublesome, he added. Learners frequently struggle with knowing whether to put the decrease or increase count statement first. Programming offers the increment (++) and decrement (-) operators, which both raise or reduce the value of the variable by 1. Leaners make mistakes when deciding whether to raise or lower a variable's value.T2 statements further reiterates the challenges encountered by T1, However T2 statement goes on to prove that the challenge in learners learning programming begins right at the very beginning when learners are being introduced to programming.

There are stages that on the introduction when you are introducing them that the most challenging is the implementation of the algorithms into source code like the syntax of the language how the language is written. You know most of them are nervous questioning themselves what is this now. When you are writing code you talk about declaration of variables the step by step or syntax of writing the codes then there will be very nervous at first but as you move on, they will get to understand better and better. Then as you move on as soon as you introduce them into functions and procedures under functions you've got ... functions procedures. We start by going to functions and procedures before we move onto user defined functions, yes that is where they also have some challenges but most of them, they will pull through.

T2 mentions items like algorithms and syntax in programming. This goes all the way to debugging which refers to understanding and fixing syntax errors, when a program is typed, typos and unintentional syntax errors are likely to occur, as well as solving algorithmic problems an algorithm is a set of guidelines used in data science to conduct out operations based on knowledge of feasible solutions to problems or address obstacles. T2 had witnessed

the implementation of the algorithms into written code, declaration of variables to be amongst the challenging topics in programming in addition to functions and procedures. However, he shifts and says that through step-by-step guidance learners eventually understand the topics that come before function.

> *Grade 10s haven't really found hard topics with grade 10 mm-hmm I* have found hard topics with grey lemons and 12. So when the programming gets, so the grade tens are fine with all the programming concepts, I start with scratch.

One the benefits teaching coding to primary school aged students would include that learners master coding easier, since in primary schools they are still developing and most like to grasp concepts easier. Since programming is a language just as most languages taught in schools such as English which taught since grade one should be provided equivalent status since it is a programming language. This links best with what T3 mentions about learners who taught themselves programming at a young age and having less difficulty mastering programming at high school age level. T3 is from a highly privileged school admitting mainly learners with high academic performance, this teacher has not noticed any challenging topics for grade 10s in programming, the reasoning is most of the learners learn to program at a very young age, she admits that at times they are ahead of T3 in terms on programming the know more than the teacher and are able to access the school's security system.

> I spend majority of my time doing coding and then obviously the subject theory, but in a term of weeks, we would spend six weeks doing coding, I finds again that the used to sitting down and passively just forward content.

Programming has always been considered challenging, mainly because introducing programming to high school and university students has posed substantial challenges and neglected to encourage the acquisition of advanced cognitive function (Sleeman, 1986). T4 agree just as T2 that it is not just functions and procedures that prove to be a challenging topic for grade 10 learners but implementing all if not most codes in their programs. This proves to be time consuming as the teacher would spend weeks trying to guide learners to implement effective and rung codes.

4.4 Differences between function and procedure

Most of the teachers are well informed on the characteristics that differentiate a function to a procedure and vice versa are their repose below is evident to this. in the sense that the two

terms are synonymous to each other by most teachers. Irrespective of the fact that functions and procedures are different are distinguished with similar yet district qualities. A function would give the receiving code the returning value. While a procedure does not yield any result to the invoking function or the code, it does return control.

A function is something that needs an intake to produce a result, therefore you must supply information to the function to obtain the intended result. A procedure, on the other hand, is essentially a routine that is assembled, and when it is performed, users simply activate that procedure within a specific element or section of code, so that it will be implemented. T1 makes it clear that a function requires data to be given to it to produce the outcome desired or expected by the programmer while the teacher describes a procedure to be an action that is executed in a program without an input.

> A function needs to return a value, but a procedure may or may not return anything at the end. So, what I normally give them as examples is that if you bake a cake, it is a procedure on how to bake a cake. So, what we say is that with the recipe, you can use it and have a cake at the end, or you can just have the recipe go through it and not necessarily make a cake at the end. So that is why a procedure may or may not return a product at the end. Then a function you are doing it, then at the end you want to see the result that is a function if you give them a function given a task to do at the end. Simple as that.

Since the above explanation utilized more abstract academic language, such language proves to be difficult as a result one of the teachers T2 as resorted to using everyday experience to demonstrate the concept more understandably to learners, using everyday language simplifies learning as it provides learners personal relevance with the concept being taught. T2 includes everyday examples to elaborate the distinction between the two concepts, the teacher demonstrate that a function should produce a result, however a procedure is more of a step-by-step process on how to carry out an action, however it holds the potential to either produce or produce an outcome. Contrary to a procedure a function always produces an outcome.

> *No, I don't know what you would call a function or a procedure. I mean, a function. I would talk to call methods and stuff like that. Oh, okay. And procedures are like, I don't know. I don't really use the term, except for like in, as in ISO procedures, we follow these procedures, you know? Okay. Um, so the terminology, isn't something I use at work. And so, I don't, and I don't use it that much in Java either.*

Another obstacle to learners' confusion of concepts when taught is due to the lack of teacher knowledge about the concept, Teacher knowledge is of paramount importance as it acts as a foundation as well as guide to learner's competency or mastery of the skill or concept that is being taught. This is evident through T3 statement as well as T4's statements. T3 recognizes the term function also known as methods and admits to having no knowledge on what a procedure is, as a resort this teacher could not differentiate the two concepts.

> the learners, I told them it's all the same things. Uh, but the difference between methods and functions would be that methods are related to an object. So, in Java, we haven't them in grade 10 as yet, so doesn't make much, they still don't know what static. Explain it to them, to them a million times. But they haven't used it. So because it hasn't been used, it's not sticking in their head.

T4 admits that the two concepts are synonymous, the only difference is that a procedure is connected to an object. And that the reason T4's learners confuse the two is due to the reason that they haven't used the concepts in practicality.

4.5 Reasons for mingling procedure and functions

All IT teachers below share similar experience regarding the reasons learners mingle functions and procedures as evident their responses below the use of mathematical concepts are very prominent in Information technology or programming in general as evident in the teacher's response. Variables serve to keep records that can be accessed, linked, and altered in computer systems. They also provide data a unique identifier, making it easier for us and the viewer to interpret the code or program. iCount is a coding instruction that instructs the computer start counting using a set of allocated integers. 3.14159 is a rational number that represents the proportion of a circle's circumference to its diameter. T1's reason to what could potentially contribute to learners mingling the use of function and procedures lies in learners' inability to differentiate the mathematical qualities that each concept (Function and Procedure) poses. T1 states:

> Mostly it would be that a function is assigned to a variable. So, you would have a variable is assigned then you would have your function. Whereas a procedure does not use variable. You just write down the procedure for example you say increase the

> iCount that is a procedure whereas for example if you use Pi, you will say R radius is equal to pi whatever so that is what they get confused, where they would assign a procedure to a variable or they would assign a function to a variable that is not there.

T1 states that the confusion is due to using a procedure where a function is required and using a function where a procedure is required. So, learner's inability to distinguish the two-concept using the mathematical characteristic that each concept possesses results to learners using the two concepts interchangeably.

> The technicalities and vocabulary, of cause vocabulary they might be familiar with it, but it is all confusing to the learners. Some of them have never experienced at all in their lives. You speak of a procedure; they might be familiar with the word, but programming wise they do not know what this procedure is you are referring to. What is a function, they know the function where there's dancing and now surprised that we are having this very same function in programming so the technicality, wording is a challenge.

T2 agrees that this is due to the specialist language as well the body of words used in programming that learners have never been exposed to throughout their lives. Instead, learners confuse these two programming concepts to everyday experience such that a function would be considered an event.

> The reason they confuse the two, theoretically wise would be the terminology that is used in the resources such as textbook. The language there is abstract, so what learners do is simply memorize the very same abstract language without understanding what exactly it means. So if they don't understand the language they automatically lack the skill to apply the theoretical knowledge of a function and a procedure in practicality.

T3 is in concurrence with T2's statement, T3 also admits learners have difficulty comprehending the academic language pertained in the textbooks which leads to learners automatically failing to implement running codes, since they failed understanding the theoretical aspect, they also fail carrying out the practical aspect of the concepts.

> I think the reasons learners confuse specifically these two concepts is that they tend to think the two are different when they are the same thing.

Brief segments of code called subroutines are used in configurable programs to execute functions. A subroutine can contain any of the pattern, option, and looping components because it is a tiny program. The main distinction between functions and subroutines is how they operate. A procedure would refer to an autonomous program unit that fulfils some specific responsibilities and is linked within a larger body of program code. A function is employed whenever a result is given to the caller routine. A subroutine is used when a desired task is needed, but no value is returned. Although the explanations of either the three constructs (subroutine, function, and procedure) may make it seem like they all refer to the same thing, there are some significant differences that are easy to miss unless you pay close attention. T4 states that the reason is that learners distinguish the two concepts when they should comprehend them as the same thing.

4.6 Is the difficulty on the theoretical or practical aspect of Information Technology?

In programming, the theoretical aspect of the content being taught and learned comprise of written information about what a particular concept means, as well as some guideline or clues on how it can be used. On the other hand, practical aspect of programming requires that learners execute the skills learnt from the written content (Theoretical aspect), the various textbooks often contain examples on how a particular concept can be executed in to running code. However, just as with any language a single concept is used in multiple ways, for example in the English language there is no fixed way on how to use a verb, as is in programming a function and procedure could be used in multiple ways. The programming language is so rigid, meaning that it works following syntax. In computer programming, syntax refers to the notion of supplying specified word sets to computers in specific sequences so that they perform what the programmer wants them to do.

Because each programming language employs distinct word sets in different sequences, each programming language has its unique syntax. Therefore, it is difficult to change language specific in programming, unlike human being language that can be used in different contexts. For example, a learner can memorize a theoretical example shown in a textbook on how to use a function or procedure that learner may only remember what they memorized the theoretical aspect, but because they have not mastered the syntactical skill of coding are most likely to get lost when writing coding of a function and or procedure. The participants

comment below suggest that the syntactical aspect of programming prove to be the most challenging for learners to comprehend. T1 commented:

> Yes, it is not in the theoretical sense, the challenge lies on the practical sense, more so on how to use a function or when rather to use a function and when to use a procedure give the question that is given to them. They lack the understanding and following the syntax code.

There is evidence from TI comments that the issue of not understanding the code is pinned on the lack of understanding of the syntax code. In addition, the difficulty lies in the practical aspect where learners become conflicted as to when to implement code for a function and when to implement code for a procedure. T4 reiterated that that understanding of the programming code lies on the correct use of the syntax. He commented,

> These learners like to write the programming code without following the syntax. On that aspect, I always tell them to follow the code syntax. The practical implementation of the problem needs the correct application of the syntax because that is where I assess whether they truly understood how these concepts work.

Teachers often evaluate learner's mastery of the skills that they have learned in theoretical aspect when learners are tasked to use the concepts in a programming, where learners execute running code using the specific programming concepts the teacher will be convinced to an extent that learners have mastered the use of the concept that was taught. However, learners' failure to implement correct use of syntax in programming concept suggests that learners do not understand the theoretical and practical content and logic on how the concept is executed, which would mean that learners neither mastered the theoretical nor the practical aspect of programming using the specific concept. T4 further reiterate the practicality of programming functions and procedures to be effective, learners are challenged to make use of syntaxes and are very crucial especially since that is where the teacher get to judge if whether learners have mastered the skill or obtained the outcome intended. On the contrary, T3 differs a bit to the above educators, she commented:

> I would say both theoretical and practical aspect because they understood the practical aspect then they effectively implement running codes. Instead, they get confused reading textbooks by the time they must do practical programming they are far worse.

T3 admits both practical and theory to be difficult for learners to comprehend, as the teacher agrees that learner's failure to comprehend the theory of functions and procedures result in the inability to program running codes. This suggests that learners fail to comprehend the written content in their textbooks, the skills that are suggested by their textbooks are not mastered by learners in the reading process which automatically suggest failure to execute running codes in the practical stage resulting in poor performance in both the theoretical as well as practical stage of learning. Additionally, T4 further elaborates on learner's challenge with the practical aspect of coding, He commented:

> Well, I'd say they find the most difficulty in the practicality not that they find theory easy but since they are able to memorize the definitions and example in theory if you give them a test on theory, they will pass well however a practical test they would find most difficult since memorization wouldn't help but logic and the skill to think.

In all forms of learning, memorization tend to dominate when route learning dominates than learning for comprehension. Memorization is an easy way to simply pass test or activities, However, in the long term it does not aid learning, as it suggests that their skills being taught were not penetrated by learners. T4 also agrees that the practicality is challenging since learners can hide the fact that they did not understand the content of functions and procedures when reading theory and memorizing, they do not have a place to hide the fact that they do not understand the logic of how the two concepts work in a program.

4.7 Teaching methods used by teachers to teach functions and procedures.

Problem solving is amongst the most enforced teaching methods in education, especially in practical related learning activities. This mostly because problems solving positions the learner to engage in critical thinking, especially in programming. Problem solving closely linked with cognitivism, it enables an individual to activate thinking, exercise and develop mental faculties. Which comprises of the ability to comprehend, recalling, application, evaluating, and creating. Most beneficial use of problem solving as a teaching method is the fact that it transfers to real life context, since coping with most of life's challenges requires that we think and therefore problem solve situations, this further transfers to the working context. Which fulfils one of the objectives of the national statements which is to prepare learners for the working environment, to teach them skills in schools that they will advance and utilize in their future career so that they are contributing citizens to the nation's

economy. The participants have provided problem solving or other forms of think associated with problem solving to be amongst the teaching methods that they resort to when teaching functions and procedures in the classroom. T1 commented:

I mostly go for problem solving and experimenting, when they are tasked with a question or activity, I give them permission to try to solve the question the way they thought best, such as using different search engines to browse the internet.

T1 confirms that he uses problem solving along with some form of discovery learning as teaching methods for functions and procedures. Discovery learning assist to make learners autonomous, learners become confident when they are able to solve problems and search for potential solutions to a problem by themselves, however, discovery learning has to be implemented with some form of guidance from the teacher. Another teaching method that teachers use is game based. Game based learning promotes learner engagement, learners are more likely to recall skills or techniques on how to implement certain programming codes because these code about functions and procedures would have been learned or practiced using a rather playful or fun teaching method. However, it is wise to note that game-based learning requires specific kinds of games specifically made for educational purposes. T2 confirms this through the following statement:

I use game-based learning, I create activities in the form of puzzles, matching the correct explanation to the key word as well as set up competitions for learners to compete on who knows the specific concepts better then reward the winner towards the end of the game.

T2 confirms that he uses games as method for learning and creating activities such as puzzles, match making which are activities that are part of problem solving and critical thinking, additionally to that, he also implements positive reinforcement, he associates learners ability to solve a problem within a task with rewards that way learners stay motivated to solve more problems in class. Using repetition as a teaching method like all learning methods has its own downfalls and uphill's. Repetition can work on the teacher's and learner's favour if it is the practice of practical work of the coding functions and procedures since individuals master practicality through repeated practice. However, if the repetition is that of route learning it will simply result to learners simply memorizing and repeating after the teacher and not attaining many skills. T4 states:

I usually use the repetition, I repeat the lesson, use previous examination papers.

T4 appears to be on the uphill side of using repetition as a teaching method for functions and procedures since she uses repetition by revising past examination papers which comprise both of practical and theoretical skills. Tasked based learning is also another effective teaching method, however, task-based learning means that the process of the learning will occur in the form of a set of questions that learners would have to answer whilst the teacher explains and guides learners towards the correct responses through the teaching process. This is useful since it is a combination of teaching and assessing learners' comprehension using a task or an activity. T3 commented:

The main one that I'm finding success in, I do not know what you would call it, but it's just making the activities and giving to the children, then I teach using the task.

T3 uses an activity, in which he conducts the lesson following a set of question. Not only is he simply teaching, but he is also able to evaluate if learners have mastered the skills taught as well as learners' ability to understand and respond to programming questions.

4.8 Remediation

Strategies on how to remedy a situation where learners tend to confuse concepts taught to them at an earlier stage tend to be simple, However, as simple as these techniques could be they are not guaranteed to aid learners, especially since teachers take the easy way out and by simply re implementing strategies that were utilized by other teachers in the past or what teachers have experience their colleagues use. What would be most beneficial is as if teachers get creative and begin to research strategies that could work best to remedy a situation where learner confuse concepts, especially because different techniques apply to different subject. A remedying technique that works for a misconception encountered by a learner in English subject wouldn't necessary aid a misconception encountered by a learner in the information technology subject. Therefore, creativity on the teacher's part would be very critical in situation where they must think and implement strategies to aid learner's confusion of certain concepts. The educators have one similar remedy technique that they use to remedy a situation where learners confuse functions and procedures. T1 commented:

> I once started out by reteaching the topic because maybe they didn't understand it the first time but what I found is just giving them more practice examples so give them more activities to do they focuses purely on functions and procedures so for example and activity that had like calculate the area the radius the diameter of shapes that

> would be a good way to use functions in order to remediate and to get them to focus on the topic more.

T1 says that the most useful way to remedy the situation is to provide another lesson on the topic, however this time the lesson will be more centred around the practicality aspect of the concepts. This is simply repetition of the same lesson, which is an effective tool to use to accommodate learners who may have fallen short on understanding the lesson the very first time around it was taught. T1 further goes on to reveal that the use of activities is the best method to access if learners have finally arrived at an understanding of the two concepts.

Due to reasons that T2 has attested to having encountered no experience with her learners struggling or mingling functions and procedures, she has therefore, provided no information with regards to the remedy techniques she would implement upon a situation where learners confuse the two concepts. T3 tends to differ on this, he commented:

> I normally give them lots of activities that they can use as practice, sometimes we even attend on weekends to help practice and understand any misconceptions that we might have experienced over the week.

Activities used to provide constructive feedback are great tool to guide learners to the skill or skills they are expected to master. Activities also aid, to utilize learners' cognitive faculties, as opposed to simply repeating a lesson. T3 further reiterates T1 ones point of providing activities that can be used to practice the implantation of code using functions and practice. Since the curriculum provides insufficient to little time for students to practice programming the teacher takes out own time on weekends for learners to further practice as well as rectify any misunderstandings. Like the above educators, T4 commented:

> Because I know that learners will encounter this problem again in future, instead of waiting for them to get to it. I repeat the content es much as I can, especially using past question papers because I know that is most likely what they are to enter their exams. Instead of

The use of past examination papers has been provided to promote memorization. However, this is a method for revision, since examination papers often repeat similar questions, this works to the advantage of learners, as they could simply encounter a question in their exam

that they have spent a lot of time practice the answer to. T4 teaches the lesson to learners repeatedly, especially using past exam papers since most question from previous years is more likely to repeat in new exam papers.

4.9 Teachers Professional Development

In the South African context, schools tend focus more on Computer Applications technology (CAT). There is a shortage of schools that offer Information technology as well as teachers who are competent in IT or programming. Another issue seems to be the fact teacher training in higher institutions seems to favour one programming language more than the other. Most higher institutions tend to put a lot of effort lecturing student teachers on Java as opposed to delph, some higher institutions do not offer any of the two programming languages at all, instead the offer CAT. Based on this information it becomes obvious that there is lack of support for programming which begins in higher learning institutions and further progresses in the workplace (teaching). T1 commented:

> *More teacher development is required, I think IT or programming in general is such a rapidly changing subject and teachers do not know everything so I think that if the department could have more teacher training development programs so that teachers go and just brush up on what they know well they might even learn something new that will help a lot.*

Teacher development programmes in programming will help meet teachers who are lacking teaching the subject halfway as well as teachers who are only competent in CAT to advance their skills in programming. T1 Hopes that they'd be programs targeted at developing skills in programming for teachers, since teachers are poorly trained in universities or not provided sufficient support and resources on how to teach programming. T2 commented:

> *I would appreciate it if the department of education would consider teaching programming IT in lower grades. Programming is a computer language just like English is considered a language and kids are taught in English since grade 1 it would make such a huge improve had the same were done with IT.*

Languages are taught at early schools' grades, this is due to reasons that learner's minds are still developing and therefore, they easily grasp information easily T2 pleads for the

government to consider and allow programming to be taught from an early age so that learners would find it less complicated to understand it as they progress through other grades and tertiary institutions. T3 would like to see more workshops on programming. She says,

> Indeed, I would appreciate it if we had some workshop on programming. There are a lot of concepts that we need to learn in programming. The challenge is that sometimes we want to learn those concepts at the sometimes teaching them to learners. We as teachers should have knowledge that is higher than that of learners.

T3 admit that she does not have much knowledge regarding programing, however it is not known to the school since she tries only tries find her way through the content that she must teach, since teaches the same thing each year it is not noticeable that she does not know the subject, she admits that knowing to teach the programing language does not in any way suggest that she is knowledgeable about it. T4 is of the opinion that there should be more use of virtual learning. He commented,

I will plead for a virtual resource that will supplement a textbook to be created, because what I have noticed that textbook is dragging and condition learners to feel overwhelmed. So, creating a virtual resource that is accessible to all learners would be great.

T4 advocates for the creation of an online tool that will be available to all learners, since textbook resources are very traditional and most like give away the message that learning is overwhelming.

4.10 Data Analysis

One of the most challenging and elusive aspects of a qualitative study, data analysis has received the least attention in the literature (Thorne 2000). The study analyses data using thematic analysis. A powerful yet flexible approach for analysing qualitative data, thematic analysis can be used in a variety of phenomenological or epistemological situations. A qualitative data analysis technique called thematic analysis involves looking through a data set to find, assess, and document recurring themes (Braun and Clarke 2006). Although it is a data description approach, selecting symbols and creating themes both involve interpretation. The capacity to use thematic analysis within a variety of theoretical and epistemological frameworks, as well as to a variety of research topic, methodologies, and data sets, makes it unique. It is vital to define the term theme in this analytical method before delving into the precise stages of thematic analysis. A theme is a consistent reaction or

interpretation of the information that supports the research topic (Braun and Clarke 2006). A theme refers to a more elusive notion necessitating additional collected information and assimilation than a category, that provides characterization and arrangement to the latent of a data collection (Nowell et al. 2017). No matter number of times a particular idea or item connected with a theme emerges in what seems like a data analysis, researchers can still find themes utilizing thematic content analysis. As a result, the following themes emerged strongly from the research:

- Lack differentiation of procedure and function
- Improper use of code syntax
- Teachers' professional growth

4.10.1 Theme1: Lack of differentiation of procedure and function

The mingling of functions and procedures is among the many misconceptions that make programming a challenge for learners to comprehend, this contributes to poor achievements by students as well as some students changing streams of subjects in schools and those at the tertiary level either drop out their modules or drop out from the institutions (Cheah, 2020). The primary challenge that contributes to learners' impairments is the learner's inability to carry out problem-solving abilities when tasked with solving programming codes (Cheah, 2020). Learners face a lot of challenges when writing programming codes that involve procedures and functions. From data presentation, there is evidence that learners confuse the two. Yet, functions simply returns and answer, there is always something in front of the function because it returns one result. Whatever is in the () is a parameter. that is the information that the function needs. that is what we are sending to the function, it is like an input to the function. it will send it back and store in a variable. The table below shows some examples of the function and the procedure.

Function	Procedure
sNum:=IntToStr(5)//change the value 5 from an interger to a string	Inc (iCount)//increase the value count by one
sNum:=InputBox('1','2','3');// Input values 1, 2, 3 in sName	showMessage ('Hello Nothando')// displays hello Nothando

sTemp:=Copy(sName,3,6)// copying 6 characters from sName starting from position 3.	Delete (sName, 4,2)// deletes two character from stored in sName from the 4th position

Clearly the function returns something to a calling statement, yet procedures do not return a value, however, the variables can be modified from inside the procedure, e.g., Delete (sName, 4,2). A function contains both value and variable arguments. The parameters are subject to the same rules as the procedure's parameters. At least one assignment operator, which assigns a value to the function name, should be included in the function.

A procedure has the following syntax.

Procedure call syntax:

<procedure name> (<call parameters list>).

A procedure should be defined in the Initiating phase before it is mentioned in the call. A procedure needs to be transcribed before it can be called. Routines are independent assertion units that can be invoked at various places throughout a script. Routines are the collective name for both functions and procedures. A function is a program that executes and returns a value. A procedure is a process that shouldn't result in a value.

The participants responses to the potential response to learners mingling functions and procedures proved to suggest a strong issue with the technical language used in defining, explaining, or distinguishing the two concepts. For instance, T1 had mentioned:

> *A function needs to return a value, but a procedure may or may not return anything at the end. So, what I normally give them as examples is that if you bake a cake, it is a procedure on how to bake a cake. So, what we say is that with the recipe, you can use it and have a cake at the end, or you can just have the recipe go through it and not necessarily make a cake at the end. So that is why a procedure may or may not return a product at the end. Then a function you are doing it, then at the end you want to see the result that is a function if you give them a function given a task to do at the end. Simple as that.*

T1 statement indicates that since the language pertained in textbooks is very technical, he resorts to incorporating examples using learner's everyday experiences. This is to assist lessen the confusion encountered by learners, however, since it is very few

teachers that implement this teaching method the mingling of function and procedures in learners remains prominent.

4.10.2 Theme 2: Improper use of code syntax

The issues of Syntaxes featured strongly from the participant. Learners either fail to use the syntax correctly when writing the code or they forget the application of that syntax. Defects in the program's grammatical structure, punctuation, and phrasing are referred to as syntax errors. Syntax mistakes frequently make a code difficult for programmers to understand, rendering them simple to spot. Compilers frequently catch the basic syntax issues, but often glitches may not always send students in the appropriate path to make the necessary corrections to the code (Hristova, Misra, Rutter, & Mercuri, 2003).

For example, students conflate the assignment operator (=) with the comparison operator (==) when they see the corresponding Program code syntaxes, such as = versus ==. Inadvertent reassignment in conditional expressions may result from this. Utilisation of == rather than equals when comparing strings results in incorrect string comparisons. .equals compares the values of two strings, whereas == simply matches the main memory of the same strings (Hristova, et al., 2003).

Incorrect counting, and/or abuse of, [], (), and Using inverted commas, square brackets, unbalanced parenthesis, or any combination of these symbols (Hristova, et al., 2003). An improper semicolon preceding an if parametric method before the if statement or preceding for or while repetition structure before the corresponding for or while loop. The program does nothing if the if condition is true, or for the duration of the for or while loop, if a semicolon is placed after the parentheses defining the if, for, or while conditions. This is legitimate code, however it's generally unpleasant and is frequently unintentionally used by new programmers (Hristova, et al., 2003). Wrong separators in for loops (using commas instead of semi-colons) Separating the initialization, testing and update clauses of a for structure with commas or other punctuation in place of semicolons (Hristova, et al., 2003). Likewise, if a bracket is used in place of parentheses Instead of using parentheses, you should use brackets to place the condition of an if statement (Hristova, et al., 2003). Using words in variable names or method names Java has 50 keywords that are reserved for specific purposes only. Thus, the names of newly created variables and functions declared by

programmers cannot contain keywords. Also, to be noted is the case-sensitivity of some programming languages like Java. The use of logical operators like >= and = can be challenging, especially for students who struggle with math. They consistently misread the signs (Hristova, et al., 2003).

Teachers must illustrate the way both the syntax of function and procedure operates. A procedure or function's name, the quantity and kind of arguments it accepts, as well as the kind of its return value in the case of a function, are all given when the procedure or function is declared. The prototype, heading, or header are additional names for this portion of the declaration. The routine or block's body, which is a block of code that is often created next, is what executes whenever the procedure or function is called.

A procedure declaration has the form:

```
procedure procedureName(parameterList); directives;
  localDeclarations;
begin
  statements
end;
```

where procedureName can be any valid identifier, statements are a list of statements that will be executed when the procedure is called, and (parameterList), directives; and localDeclarations; are optional. Like a procedure declaration, a function declaration also provides a return type and return value.
The form of a function declaration is:

```
function functionName(parameterList): returnType; directives.
  localDeclarations;
begin
  statements
end;
```

Where functionName may be any valid identifier, returnType can be any type identifier, statements can be any sequence of statements that run when the function is called, and (parameterList), directives; and localDeclarations; are optional. Participants reiterated that that understanding of the programming code lies on the correct use of the syntax. It was

noted that their comment emphasised that learners need to be shown repeatedly on how to write a function or procedural statement following the syntax.

4.10.3 Theme 3: Teachers' professional growth

Participants strongly advocated the need for professional growth. Professional development is an essential component of retaining high quality teachers. Although professional development opportunities have been abundant for decades, a closer look at standards and criteria for effective professional development is essential to increase the possibility of programming opportunities that will truly impact teacher behaviours (Good & Weaver, 2003). Teachers, like students, are learners, and to continually develop and encourage teacher effectiveness throughout the course of an individual's career, teachers must stay current in educational theory and practice (Good & Weaver, 2003). Thus, it seems to be a logical assumption that teachers must continue to learn about their craft and content, making quality professional development opportunities essential. As evident on the data presentation teachers have very different yet common instructional methodologies, these range from the incorporation certain teaching and learning theories, using examples in reference to everyday experience, repeat the lesson content while placing emphasis on the practical aspect. To some extent, these approaches are effective since teachers lack training on the specific procedures that are most effective for the specific subjects that they are teaching (in this case, Information technology). As a result, teachers advocate for teacher development programs and the introduction of programming to students in the early primary school years.

4.11 Discussion

The data presented has proven that prominent contributor to learner's confusion of the functions and procedures in programming is due to the high level of technical language that is used in the content accessible to learners through textbooks as well as teacher's teaching methodologies and instruction. The teaching techniques used by teachers are general and are not in accordance with the specific subject that teachers are teaching, the teaching methodologies used such as repetition of the same lesson does not promote meaningful learning instead it furthers the agenda for route learning. Repetition does not activate thinking nor does it excise and develop cognitive faculties.

The LCT uncovers what it takes to succeed or the rules of the game, since technical language is an issue in learners confusing the two concepts. What could help learners succeed in this cause is the use of semantic waves to unpack the technicality that exist. The participants have admitted to learners' inability to differentiate the two concepts due to the abstract language in the theoretical aspect of learning which contributes to learners' automatic failure in implementing running codes on the practical stage of learning.

This is regarded as a strong Semantic Density (SD+) the principle that governs complexity since it is strong, it suggests that the level of complexity and used of concept in the content being learned by learners is high. In this case the Semantic Gravity is weak (SG-), since the content is being taught is not as context dependent and the use of abstract language is more dominant in the learning resources that learners interact with. This means that there are few cases were learners are taught function and procedures using example that relate with everyday experience or using everyday language.

Semantic waves would then lessen the misconception of functions and procedures by indicating the knowledge being taught (abstract concept), then moving on to (unpacking) the concepts in steps by providing a series of more specific (concrete) examples and then link the examples back to the abstract idea (repacking). According to Dlamini and Dewa (2022), information technology educators when applying semantic wave of the LCT can move through the process of creating programming concepts and pieces using more and less details as necessary. Therefore, it is crucial to pack, unpack, and pack again while switching between various programming ideas and notions. Explain the usage of letters and what sort of data type they will hold, for example, while introducing the principles of variables in programming. Assign data to stated variables and show the variables' contents.

After that, learners might be assigned their own assignments to practice the newly learnt concept. They gain mastery of the concepts by practising more often. Teachers are urged to apply the semantic wave of LCT by sketching out the learning outcomes, such as what information will variables in programming store. Thus, a LCT is seen as a potential framework that build knowledge about education and society that information technology teacher can use to reinforce their teaching practice.

4.12 Conclusion

It appeared from the experience of the participants that functions and procedures are a challenge to teach and learn. The different factors that contribute to learners mingling

functions and procedures have been indicated, ranging from the teaching methods used, learners' failure to differentiate the two concepts and follow correct syntax to a gap in teachers' development. This chapter has focused on analysing and interpreting the collected data.

Chapter Five: Findings, Conclusion and Recommendation

5.1 Introduction

The previous chapter focused on data analysis and interpretation to produce the findings of this study. This chapter provides an overview of the study and its conclusions. Further to this, it will summarise the research findings and provide recommendations Enhancing the teaching and learning of programming functions and procedures using Legitimate Code Theory. Finally, suggestions for future research will be provided.

5.2 The findings

The following is how the study answered to the research questions:

1. ***What methods are IT teachers using in teaching functions and procedures in programming?***

Information technology teachers indicated that they use standard or rather basic teaching methods to teach functions and procedures in programming. The methods indicated comprised of problem solving and experimenting, where the teachers would embed some form of discovery learning by enabling learners to solve tasks or the problem at hand in a manner they considered or found effective, such as using the internet to search for solutions to the task and continuously experimenting the effectiveness of the examples that the internet provides as potential solutions to the problem or task at hand. The use of game-based learning, where activities are created in the form of puzzles, matching the correct explanation to the specific concepts as well as competitions where the winner is awarded a reward based on their level of mastery for functions and procedures concepts. Another teaching method teachers have attested to using was repetition of the same lesson on function and procedures as well as revising with previous question papers and simply constructing activities and allowing learners to complete those activities. by the teacher These methods I refer as basic due to reasons that they are simply methods that most teachers irrespective of the subject they teach resort to in teaching content.

2. ***How do the methods used in teaching functions and procedures in programming make learners mingle the two concepts?***

These teaching methods are constructed with the aim to simply teach and not for learners to be taught and master the lessons content with an understanding. Although teachers have

indicated to repeat the same lesson, use game-based learning, and construct activities, teachers continue to teach content that is dominated by technical language. This includes content that is taught using abstract concepts, concepts that are taught using a language that is difficult for learners to understand, also known as high sematic gravity (SG+) content dominated by conceptual language, is more complex and uses no to little everyday language or examples relating to every experience. Also, these methods are a standard procedure in teaching, IT teachers do not put in efforts to research teaching methodologies that are most effective in teaching the specific concepts of functions and procedures.

3. How can the Semantic wave of Legitimate Code Theory help teachers to model the concepts of functions and procedures in programming?

Since the content being taught is not as context dependent and the use of abstract language is more dominant in the learning resources that learners interact with. This means that there are few cases were learners are taught function and procedures using example that relate with everyday experience or using everyday language. Semantic waves would then lessen the misconception of functions and procedures by indicating the knowledge being taught (abstract concept), then moving on to (unpacking) the concepts in steps by providing a series of more specific (concrete) examples and then link the examples back to the abstract idea (repacking).

Main Research Question

What makes learners in grade 10 mingle with the use of functions and procedures in Information Technology (IT) programming?

It has become apparent that teacher's poor selection of teaching methodologies that are targeted at improving learners understand or mastery of the functions and procedure in programming concepts as well as the dominance of technical language used when teaching learners as well as contained in learner's learning resources such as textbooks are the two main reasons learners mingle the notion of functions and procedures.

5.3 Recommendations

No study can cover all aspects of a given phenomenon. Therefore, the following suggestions are offered for future exploration. Recommendations are provided in the following sections:

• Teachers.

• The Department of Education (DoE).

5.3.1 Teachers

The study found that teachers use of inappropriate teaching methods as well the dominance of technical language used in the teaching process as well as contained learning resources were reasons learners mingled functions and procedures. Teachers should be more selective on teaching methodologies they select to teach programming especially the concepts of functions and procedures. Teachers should conduct through research on the teaching methods that are most suitable to teach the two concepts (functions and procedures) in programming. Also, teachers should be aware of the technicality of the language used in programming and attend to this by using the semantic waves of Legitimate Code Theory mostly focusing on the concepts of Semantic Gravity since that will assist teachers to unpack the technical language, by breaking down difficult language and using every example to explain the concept in simpler terms and then repacking by relating the concepts' explanation and examples to its original meaning.

5.3.2 Department of Education DoE

The department of education is advised to implement teacher development programs that will assist in the growth of teacher's skills in mastery the appreciate teaching of concepts in programming, as it has been found in this study that some teachers struggle to understand the subject concepts themselves which automatically results and poor teaching and learners' failure to execute the required or appreciate skills.

5.3.3 Suggestion for further research

Although the study has reached its goal to find out way of enhancing the teaching of functions and procedures using Legitimate Code Theory. There is still space for further research to uncover what the specific appropriate teaching methodologies are for teaching programming concepts such as functions and procedures.

The extensive use of LCT to unpack not just functions and processes, but all ideas in programming, from how teachers organize their teaching instructions to how teachers design questions in their activities, assessments, and assignments, should be practiced.

5.4 Conclusion

In conclusion, the study highlighted the underlying challenges that contribute to learner's mingling of functions and procedures in programming. This was done to understand how the

LCT can be used to enhance the teaching and learning of functions and procedures, The evidence presented showed that inappropriate selection of teaching methodologies by teachers to teach functions and procedures as well as the use of technical language are the main contributors to learner's mingling functions and procedures. In that case, semantic waves of Legitimate Code Theory can be used as way forward to unpack abstract concepts by breaking difficult language down through explanation using everyday language and everyday experience examples then repacking by relating the examples to original meaning.

References

Acharya, A. S., Prakash, A., Saxena, P., & Nigam, A. (2013). Sampling: Why and how of it. *Indian Journal of Medical Specialties*, *4*(2), 330-333.

Aliard, R. W. (1970). Population structure and sampling methods. *Population structure and sampling methods.*

Alvarez, A., & Scott, T. A. (2010). Using student surveys in determining the difficulty of programming assignments. *Journal of Computing Sciences in Colleges*, *26*(2), 157-163.

Armoni, M., Benaya, T., Ginat, D., & Zur, E. (2010). Didactics of introduction to computer science in high school. In *Teaching Fundamentals Concepts of Informatics: 4th International Conference on Informatics in Secondary Schools-Evolution and Perspectives, ISSEP 2010, Zurich, Switzerland, January 13-15, 2010. Proceedings 4* (pp. 36-48). Springer Berlin Heidelberg.

Asongu, S. A., Orim, S. M. I., & Nting, R. T. (2019). Inequality, information technology and inclusive education in sub-Saharan Africa. *Technological Forecasting and Social Change*, *146*, 380-389.

Aspers, P., & Corte, U. (2019). What is Qualitative in Qualitative Research. *Qualitative Sociology*, *42*(2), 139–160. https://doi.org/10.1007/s11133-019-9413-7

Baker, M. J. (2000). Selecting a research methodology. *The marketing review*, *1*(3), 373-397.

Baxter, P., & Jack, S. (2008). Qualitative case study methodology: Study design and implementation for novice researchers. *The qualitative report*, *13*(4), 544-559.

Benner, C. (2006). 'South Africa on-call': Information technology and labour market restructuring in South African call centres. *Regional studies*, *40*(9), 1025-1040.

Blackie, M. A. (2014). Creating semantic waves: Using Legitimation Code Theory as a tool to aid the teaching of chemistry. *Chemistry Education Research and Practice*, *15*(4), 462-469.

Blackwell, A. F. (2002, June). What is programming?. In *PPIG* (p. 20).

Bunniss, S., & Kelly, D. R. (2010). Research paradigms in medical education research. *Medical education*, *44*(4), 358-366.

Byrne, P., & Lyons, G. (2001, June). The effect of student attributes on success in programming. In *Proceedings of the 6th annual conference on Innovation and technology in computer science education* (pp. 49-52).

Cheah, C. S. (2020). Factors contributing to the difficulties in teaching and learning of computer programming: A literature review. *Contemporary Educational Technology*, *12*(2), ep272.

Chiang, T. H. C. (2018). Analysis of learning behavior in a flipped programing classroom adopting problem-solving strategies. In *Learning Analytics* (pp. 47-60). Routledge.

Clements, M., & Braman, J. (2021, July). Exploring Factors of an Inclusive Textbook Access Program in Computer Technology Courses. In *Social Computing and Social Media: Applications in Marketing, Learning, and Health: 13th International Conference, SCSM 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings, Part II* (pp. 187-198). Cham: Springer International Publishing.

Cohen, L., Manion, L., & Morrison, K. (2002). *Research methods in education*. routledge.

Corney, M., Teague, D., & Thomas, R. (2010). Engaging students in programming. In *Proceedings of the twelfth Australasian computing education conference* (pp. 63-72). Australian Computer Society.

Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.
Department of Education. 2003. National Curriculum Statement. Grades 10-12 (General). Information Technology, 1-63.

Dlamini, R., & Dewa, A. (2022). Unplugged Teaching: Deepening Information Technology Learning. *Open Journal of Social Sciences*, *10*(4), 476-486.

Felleisen, M., Findler, R. B., Flatt, M., & Krishnamurthi, S. (2004). The TeachScheme! project: Computing and programming for every student. *Computer Science Education*, *14*(1), 55-77.

Forišek, M. (2010, January). The difficulty of programming contests increases. In *International Conference on Informatics in Secondary Schools-Evolution and Perspectives* (pp. 72-85). Berlin/Heidelberg, Germany: Springer.

Franklin, D., Hill, C., Dwyer, H. A., Hansen, A. K., Iveland, A., & Harlow, D. B. (2016, February). Initialization in scratch: Seeking knowledge transfer. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 217-222).

Gomes, A., Carmo, L., Bigotte, E., & Mendes, A. (2006, September). Mathematics and programming problem solving. In *3rd e-learning conference–computer science education* (pp. 1-5).

Good, J. M., & Weaver, A. (2003). Creating learning communities to meet teachers' needs in professional development. *Journal of In-Service Education*, *29*(3), 439-450.

Gorder, K. L. (1980). Understanding School Knowledge: A Critical Appraisal of Basil Bernstein and Pierre Bourdieu. *Educational Theory*, *30*(4), 335-46.

Gupta, A., & Gupta, N. (2022). *Research methodology*. SBPD Publications.

Harker, R., & May, S. A. (1993). Code and habitus: Comparing the accounts of Bernstein and Bourdieu. *British journal of sociology of education*, *14*(2), 169-178.

Havenga, M., & Mentz, E. (2009, June). The school subject information technology: a South African perspective. In *Proceedings of the 2009 Annual Conference of the Southern African Computer Lecturers' Association* (pp. 76-80).

He, W., Zhang, Z. J., & Li, W. (2021). Information technology solutions, challenges, and suggestions for tackling the COVID-19 pandemic. *International journal of information management*, *57*, 102287.

Holton, G. (2009). Relatively ethical: A comparison of linguistic research paradigms in Alaska and Indonesia.

How, A. (2017). *Critical theory*. Bloomsbury Publishing.

Hristova, M., Misra, A., Rutter, M., & Mercuri, R. (2003). Identifying and correcting Java programming errors for introductory computer science students. *ACM SIGCSE Bulletin*, *35*(1), 153-156.

Hua, Z. (2015). Identifying research paradigms. *Research methods in intercultural communication: A practical guide*, 1-22.

Idrees, M., & Aslam, F. (2022). A Comprehensive Survey and Analysis of Diverse Visual Programming Languages.

Intisar, C. M., & Watanobe, Y. (2018, November). Cluster analysis to estimate the difficulty of programming problems. In *Proceedings of the 3rd International Conference on Applications in Information Technology* (pp. 23-28).

Jenkins, T. (2002, August). On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences* (Vol. 4, No. 2002, pp. 53-58).

Joorabchi, A., English, M., & Mahdi, A. E. (2016). Text mining stackoverflow: An insight into challenges and subject-related difficulties faced by computer science learners. *Journal of Enterprise Information Management*, *29*(2), 255-275.

measurement. *MIER Journal of Educational Studies Trends and Practices*, 46-57.

Khaldi, K. (2017). Quantitative, qualitative or mixed research: which research paradigm to use?. *Journal of Educational and Social Research*, *7*(2), 15.

Kivunja, C., & Kuyini, A. B. (2017). Understanding and applying research paradigms in educational contexts. *International Journal of higher education*, *6*(5), 26-41.

Koorsse, M., Cilliers, C., & Calitz, A. (2015). Programming assistance tools to support the learning of IT programming in South African secondary schools. *Computers & Education*, *82*, 162-178.

Kothari, C. R. (2004). *Research methodology: Methods and techniques*. New Age International.

Krauss, S. E. (2005). Research paradigms and meaning making: A primer. *The qualitative report*, *10*(4), 758-770.

Ledgard, H. F. (1971). Ten mini-languages: A study of topical issues in programming languages. *ACM Computing Surveys (CSUR)*, *3*(3), 115-146.

Luxton-Reilly, A. (2016, July). Learning to program is easy. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 284-289).

Mannila, L., Peltomäki, M., & Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer science education*, *16*(3), 211-227.

Maton, K., (2014). Knowledge-building. Educational studies in Legitimation Code Theory. Routledge.

Maton, K., Hood, S., & Shay, S. (2016). Knowledge-building.

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2010, August). Learning computer science concepts with scratch. In *Proceedings of the Sixth international workshop on Computing education research* (pp. 69-76).

Morgan, G. A., & Harmon, R. J. (2001). Data collection techniques. *Journal-American Academy Of Child And Adolescent Psychiatry*, *40*(8), 973-976.

Pardaboyevich, R. F., Abdunazirovich, U. S., & Saydullayevich, S. Q. (2020). Teaching computer science at school-current challenges and prospects. *JournalNX*, *6*(11), 217-221.

Pathak, V., Jena, B., & Kalra, S. (2013). Qualitative research. *Perspectives in clinical research*, *4*(3).

Patton, B. (2020). An analysis of factors that may Influence student satisfaction in computer programming courses at an online Midwestern University.

Patton, M. Q. (2005). Qualitative research. *Encyclopedia of statistics in behavioral science*.

Python programming: an introduction to computer science. Franklin, Beedle & Associates, Inc..

Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, *18*(1), 1-24

Rahi, S. (2017). Research design and methods: A systematic review of research paradigms, sampling issues and instruments development. *International Journal of Economics & Management Sciences*, *6*(2), 1-5.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, *13*(2), 137-172.

Rusznyak, L. Using semantic pathways to reveal the 'depth 'of pre-service teachers' reflections.

Santos, S. C., Tedesco, P. A., Borba, M., & Brito, M. (2020). Innovative approaches in teaching programming: A systematic literature review. In *Proceedings of the 12th International Conference on Computer Supported Education* (Vol. 1, pp. 205-214).

Scheepers, H., & de Villiers, C. (2000). Teaching of a computer literacy course in South Africa: A case study using traditional and co-operative learning. *Information Technology for Development*, *9*(3-4), 175-187.

Shinebourne, P. (2011). The Theoretical Underpinnings of Interpretative Phenomenological Analysis (IPA). *Existential Analysis: Journal of the Society for Existential Analysis*, *22*(1).

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of business research*, *104*, 333-339.

Tucker, A. B. (2010). K-12 computer science: Aspirations, realities, and challenges. In *Teaching Fundamentals Concepts of Informatics: 4th International Conference on Informatics in Secondary Schools-Evolution and Perspectives, ISSEP 2010, Zurich, Switzerland, January 13-15, 2010. Proceedings 4* (pp. 22-34). Springer Berlin Heidelberg.

Vishnevsky, T., & Beanlands, H. (2004). Qualitative research. *Nephrology Nursing Journal*, *31*(2), 234.

Walker, D. H. (1997). Choosing an appropriate research methodology. *Construction management and economics*, *15*(2), 149-159.

Walsham, G. (2006). Doing interpretive research. *European journal of information systems*, *15*(3), 320-330.

Xu, S., & Rajlich, V. (2004, August). Cognitive process during program debugging. In *Proceedings of the Third IEEE International Conference on Cognitive Informatics, 2004.* (pp. 176-182). IEEE.\

Zhang, J. (2021, April). Teaching Design and Implementation of Design Mode in the Course of Java Programming in Higher Vocational Colleges. In *Journal of Physics: Conference Series* (Vol. 1856, No. 1, p. 012039). IOP Publishing.

Appendix 1

SCHOOL OF EDUCATION ETHICS COMMITTEE

CONSTITUTED UNDER THE UNIVERSITY HUMAN RESEARCH ETHICS COMMITTEE (NONMEDICAL)

CLEARANCE CERTIFICATE PROTOCOL NUMBER:
2022ECE039M

PROJECT TITLE Enhancing the teaching and learning of programming functions and procedures using Legitimate Code Theory.

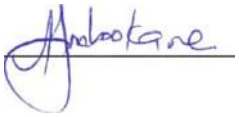
INVESTIGATOR	Wadi Nothando Truely
SCHOOL/DEPARTMENT OF INVESTIGATOR	**WSOE**
DATE CONSIDERED	22 July 2022
DECISION OF THE COMMITTEE	Approved unconditionally.

RISK LEVEL No Risk

EXPIRY DATE Date of submission of the Research Report

ISSUE DATE OF CERTIFICATE CHAIRPERSON

27 St Andrews Road, Parktown, Johannesburg, 2193 | Pvt Bag 3, WITS, 2050 | **T** +27 11 717 3007 | www.wits.ac.za/education

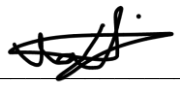
Dr. Batseba Mofolo-Mbokane 

cc: Dr Alton Dewa

DECLARATION OF INVESTIGATOR

To be completed in duplicate and **ONE COPY** returned to the Chairperson of the School/Department ethics committee.

I fully understand the conditions under which I am are authorized to carry out the abovementioned research and I guarantee to ensure compliance with these conditions. Should any departure to be contemplated from the research procedure as approved I/we undertake to resubmit the protocol to the Committee.



Signature

29 / 07 /2022

Date

Appendix 2

GAUTENG PROVINCE

Department: Education
REPUBLIC OF SOUTH AFRICA

814141112

GDE RESEARCH APPROVAL LETTER

Date:	27 July 2022
Validity of Research Approval:	08 February 2022- 30 September 2022 20221282
Name of Researcher:	Wadi N.T
Address of Researcher:	Stand no 221
	Chrissiesmeer
	Mpumalanga
Telephone Number:	078 414 2566
Email address:	1741809@students.wits.ac.za
Research Topic:	Teaching programming functions and procedures using Legitimate Code Theory Secondary Schools in Gauteng.
Type of qualification	Master of Education
Number and type of schools:	5 Secondary Schools
District/s/HO	Johannesburg East

Re: Approval in Respect of Request to Conduct Research

This letter serves to indicate that approval is hereby granted to the above-mentioned researcher to proceed with research in respect of the study indicated above. The onus rests with the researcher to negotiate appropriate and relevant time schedules with the school/s and/or offices involved to conduct the research. A separate copy of this letter must be presented to both the school (both Principal and SOB) and the District/Head Office Senior Manager confirming that permission "has been granted for the research to be conducted.

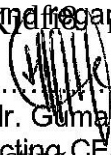
The following conditions apply research. The researcher may proceed with the above study subject to the con ions listed below are met. Approval may be withdrawn should any of the conditions listed below be flouted:

Office of the Director: Education Research and Knowledge Management
7th Floor, 17 Simmonds Street, Johannesburg, 2001
Tel: (011) 355 0488
Email: Faith.Tshabalala@gauteng.gov.za
Website: www.education.gpggov.za

1. The letter would indicate that the said researcher/s has/have been granted permission from the Gauteng Department of Education to conduct the research study.
2. The District/Head Office Senior Manager/s must be approached separately, and in writing, for permission to involve District/Head Office Officials in the project.
3. Because of the relaxation of COWD 19 regulations researchers can collect data online, telephonically, physically access schools, or may make arrangements for Zoom with the school Principal. Requests for such arrangements should be submitted to the GDE Education Research and Knowledge Management directorate.
4. The Researchers are advised to wear a mask at all times, Social distance at all times, Provide a vaccination certificate or negative COVID„19 test, not older than 72 hours, and Sanitise frequently.
5. A copy of this letter must be forwarded to the school principal and the chairperson of the School Governing Body (SGB) that would indicate that the researcher/s has been granted permission from the Gauteng Department of Education to conduct the research study.
6. A letter/document that outlines the purpose of the research and the anticipated outcomes of such research must be made available to the principals, SGBs, and District/Head Office Senior Managers of the schools and districts/offices concerned, respectively.
7. The Researcher will make every effort to obtain the goodwill and cooperation of all the GDE officials, principals, and chairpersons of the SGBs, teachers, and learners involved. Persons who offer their cooperation will not receive additional remuneration from the Department while those that opt not to participate will not be penalised in any way.
8. Research may only be conducted after school hours so that the normal school program is not interrupted. The Principal (if at a school) and/or Director (if at a district/head office) must be consulted about an appropriate time when the researcher/s may carry out their research at the sites that they manage.
9. Research may only commence from the second week of February and must be concluded before the beginning of the last quarter of the academic year. If incomplete, an amended Research Approval letter may be requested to conduct research in the following year.
10. Items 6 and 7 will not apply to any research effort being undertaken on behalf of the GDE. Such research will have been commissioned and be paid for by the Gauteng Department of Education.
11. It is the researcher's responsibility to obtain written parental consent of all learners that are expected to participate in the study.
12. The researcher is responsible for supplying and utilising his/her research resources, such as stationery, photocopies, transport, faxes, and telephones, and should not depend on the goodwill of the institutions and/or the offices visited for supplying such resources.
13. The names of the GDE officials, schools, principals, parents, teachers, and learners that participate In the study may not appear in the research report without the written consent of each of these individuals and/or organisations.
14. On completion of the study, the researcher/s must supply the Director: Knowledge Management & Research with one Hard Cover bound and an electronic copy of the research.
15. The researcher may be expected to provide short presentations on the purpose, findings, and recommendations of his/her research to both GDE officials and the schools concerned.
16. Should the researcher have been involved with research at a school and/or a district/head office level, the Director concerned must also be supplied with a summary of the purpose, findings, and recommendations of the research study.

The Gauteng Department of Education wishes you well in this important undertaking and looks forward to examining the findings of your research study.

Kind regards



Mr. Gumani Mukatuni
Acting CES: Education Research and Knowledge Management

DATE: 27/07/2022

Office of the Director: Education Research and Knowledge Management
7th Floor, 17 Simmonds Street, Johannesburg. 2001
Tel: (011) 355 0488
Email: Faith.Tshabalala@gauteng.gov.za
Website: www.education.gpg.gov.za

Declaration of Original Work

I, (Name and surname) NOTHANDO WADI

Student number: **1741809**

Know and accept that plagiarism (i.e., to use another's work and to pretend that it is one's own) is dishonest.

Please confirm the following:

✓	I declare that the assignment entitled: **Enhancing the teaching and learning of programming functions and procedures using Legitimate Code Theory.** _____ And handed in on the date below is my own work.
✓	I have acknowledged all direct quotations and paraphrased ideas.
✓	I have provided a complete, alphabetised reference list, as required by the APA method of referencing (described in the Referencing Handbook).
✓	I have not allowed, and will not allow anyone to copy my work with the intention of passing it off as his or her own work.
✓	I understand that the University of the Witwatersrand will take disciplinary action against me if evidence suggests that this is not my own unaided work or that I failed to acknowledge the source of the ideas or words in my writing

Signed: **N.T WADI**

Date: **28 February 2023**