AN INTEGRATED DEVELOPMENT ENVIRONMENT ON PROBLEM-SOLVING FOR C PROGRAMMING FUNDAMENTALS

NOR FARAHWAHIDA BINTI MOHD NOOR

05-4506832 pustaka.upsi.edu.my Perpustakaan Tuanku Bainun Kampus Sultan Abdul Jalil Shah PustakaTBainun ptbupsi

SULTAN IDRIS EDUCATION UNIVERSITY

2022

05-4506832 pustaka.upsi.edu.my Perpustakaan Tuanku Bainun Kampus Sultan Abdul Jalil Shah PustakaTBainun ptbupsi

AN INTEGRATED DEVELOPMENT ENVIRONMENT ON PROBLEM-SOLVING FOR C PROGRAMMING FUNDAMENTALS

NOR FARAHWAHIDA BINTI MOHD NOOR

05-4506832 pustaka.upsi.edu.my Perpustakaan Tuanku Bainun Kampus Sultan Abdul Jalil Shah PustakaTBainun ptbupsi

DISSERTATION PRESENTED TO QUALIFY FOR A MASTER'S IN SCIENCE (RESEARCH MODE)

FACULTY OF ART, COMPUTING AND CREATIVE INDUSTRY
SULTAN IDRIS EDUCATION UNIVERSITY

2022

05-4506832 pustaka.upsi.edu.my Perpustakaan Tuanku Bainun Kampus Sultan Abdul Jalil Shah PustakaTBainun ptbupsi

SULTAN IDRIS EDUCATION UNIVERSITY

Sila Tanda (√)

Kertas Projek	
Sarjana Penyelidikan	√
Sarjana Penyelidikan Dan Kerja Kursus	
Doktor Falsafah	

INSTITUT PENGAJIAN SISWAZAH
PERAKUAN KEASLIAN PENULISAN

Perakuan ini telah dibuat pada **15** (hari bulan) **12** (bulan) 20**22**.

i. Perakuan pelajar:

Saya, **NOR FARAHWAHIDA BINTI MOHD NOOR, M20201000081, FAKULTI SENI, KOMPUTERAN & INDUSTRI KREATIF** dengan ini mengaku bahawa disertasi/tesis yang bertajuk **AN INTEGRATED DEVELOPMENT ENVIRONMENT ON PROBLEM-SOLVING FOR C PROGRAMMING FUNDAMENTALS** adalah hasil kerja saya sendiri. Saya tidak memplagiat dan apa-apa penggunaan mana-mana hasil kerja yang mengandungi hak cipta telah dilakukan secara urusan yang wajar dan bagi maksud yang dibenarkan dan apa-apa petikan, ekstrak, rujukan atau pengeluaran semula daripada atau kepada mana-mana hasil kerja yang mengandungi hak cipta telah dinyatakan dengan sejelasnya dan secukupnya

Tandatangan pelajar

ii. Perakuan Penyelia:

Saya, **PROFESOR MADYA DR. ASLINA BINTI SAAD** dengan ini mengesahkan bahawa hasil kerja pelajar yang bertajuk **AN INTEGRATED DEVELOPMENT ENVIRONMENT ON PROBLEM-SOLVING FOR C PROGRAMMING FUNDAMENTALS** dihasilkan oleh pelajar seperti nama di atas, dan telah diserahkan kepada Institut Pengajian Siswazah bagi memenuhi sepenuhnya syarat untuk memperoleh Ijazah **MASTER'S IN SCIENCE (SOFTWARE ENGINEERING)**.

15/12/2022

Tarikh

Tandatangan Penyelia

UNIVERSITI PENDIDIKAN SULTAN IDRIS
SULTAN IDRIS EDUCATION UNIVERSITY

INSTITUT PENGAJIAN SISWAZAH /
INSTITUTE OF GRADUATE STUDIES

BORANG PENGESAHAN PENYERAHAN TESIS/DISERTASI/LAPORAN KERTAS PROJEK
DECLARATION OF THESIS/DISSERTATION/PROJECT PAPER FORM

Tajuk / *Title*: AN INTEGRATED DEVELOPMENT ENVIRONMENT ON PROBLEM-SOLVING FOR C PROGRAMMING FUNDAMENTALS

No. Matrik /*Matric's No.*: M20201000081

Saya / *I* : NOR FARAHWAHIDA BINTI MOHD NOOR

(Nama pelajar / *Student's Name*)

mengaku membenarkan Tesis/Disertasi/Laporan Kertas Projek (Kedoktoran/Sarjana)* ini disimpan di Universiti Pendidikan Sultan Idris (Perpustakaan Tuanku Bainun) dengan syarat-syarat kegunaan seperti berikut:-
acknowledged that Universiti Pendidikan Sultan Idris (Tuanku Bainun Library) reserves the right as follows:-

1. Tesis/Disertasi/Laporan Kertas Projek ini adalah hak milik UPSI.
   The thesis is the property of Universiti Pendidikan Sultan Idris
2. Perpustakaan Tuanku Bainun dibenarkan membuat salinan untuk tujuan rujukan dan penyelidikan.
   Tuanku Bainun Library has the right to make copies for the purpose of reference and research.
3. Perpustakaan dibenarkan membuat salinan Tesis/Disertasi ini sebagai bahan pertukaran antara Institusi Pengajian Tinggi.
   The Library has the right to make copies of the thesis for academic exchange.
4. Sila tandakan (√) bagi pilihan kategori di bawah / *Please tick (√) for category below:-*

☐	**SULIT/*CONFIDENTIAL***	Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub dalam Akta Rahsia Rasmi 1972. / *Contains confidential information under the Official Secret Act 1972*
☐	**TERHAD/*RESTRICTED***	Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan ini dijalankan. / *Contains restircted information as specified by the organization where research was done.*
☑	**TIDAK TERHAD / *OPEN ACCESS***	

(Tandatangan Pelajar/ Signature)

(Tandatangan Penyelia / *Signature of Supervisor*)
& (Nama & Cop Rasmi / *Name & Official Stamp*)

Tarikh: 15/12/2022

Catatan: Jika Tesis/Disertasi ini **SULIT @ TERHAD**, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh laporan ini perlu dikelaskan sebagai **SULIT** dan **TERHAD**.

Notes: If the thesis is CONFIDENTAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction.

ACKNOWLEDGEMENT

Firstly, my deepest thank to my supervisor, Associate Professor Dr Aslina Binti Saad, my co-supervisor Associate Professor Ts. Dr Abu Bakar bin Ibrahim, and all lecturers in UPSI for their continuous effort, guidance, and motivation for me in accomplishing this study. I would like to express my gratitude to the Education Sponsorship Division, the Ministry of Education (MOE), and The Government of Malaysia for the financial support through scholarships and assistance toward this study. I would also like to thank numerous people involved in this study; the expert lecturers in Programming Fundamentals teaching and learning, and software development. My appreciation is also to UPSI Post Graduate Institute, and my friends in UPSI Postgraduate Association for their motivation, criticism, and various support. Last but not least, I would like to thank my family, especially my husband, parents, children, colleagues, and friends for their unconditional support and contribution to this study.

ABSTRACT

The industrial revolution is now in great need of more skilful programmers with problem-solving skills. However, the learning process for programming is very challenging due to its complexity and limited educational application that covers both problem-solving and programming environments. This study aims to develop an introductory Integrated Development Environment (IDE) application (C-SOLVIS) and evaluate its usability. It integrates both environments for problem-solving and program development for the C language. The purpose is to guide the users in problem-solving and help in writing a C program. This study used a mix-method approach, in which qualitative methods were conducted during the requirement planning phase through a literature review supported by semi-structured interviews, document reviews, and content validation by seven expert programming lecturers. Meanwhile, a quantitative method to evaluate the application's usability among the same lecturers was conducted using the System Usability Scale (SUS) instrument to obtain its mean score. The application's development process has employed Rapid Application Development (RAD) Model in which application design has been accomplished by iterative prototyping process, followed by application construction. The study has discovered suitable techniques and designs for the problem-solving and program development environment. In the problem-solving environment, Computational Thinking (CT) concepts have been applied and supported by Input-Proses-Output (IPO) Model through Scientific Instructions and Inquiries. Meanwhile, the program development environment features frame-based programming through Code Patterns. The C-SOLVIS has achieved a SUS mean score of 86.07 which is interpreted by SUS as an A grade, indicating C-SOLVIS as a highly usable application for the teaching and learning of an introductory programming course. In conclusion, C-SOLVIS could facilitate the teaching and learning of C programming fundamentals effectively. The implication is that the development process of C-SOLVIS can be used as a guideline for educational software development, especially in the application of programming education.

PERSEKITARAN PEMBANGUNAN BERSEPADU BAGI PENYELESAIAN MASALAH UNTUK ASAS PENGATURCARAAN C

ABSTRAK

Revolusi perindustrian kini amat memerlukan lebih ramai pakar pengaturcaraan dengan kebolehan menyelesaikan masalah. Walau bagaimanapun, pembelajaran pengaturcaraan adalah sangat mencabar kerana tahap kerumitan dan keterbatasan aplikasi pendidikan yang merangkumi kedua-dua persekitaran penyelesaian masalah dan pengaturcaraan. Kajian ini bertujuan membangunkan sebuah aplikasi persekitaran pembangunan bersepadu (IDE) (C-SOLVIS) serta menilai kebolehgunaannya. Ia mengintegrasikan kedua-dua persekitaran penyelesaian masalah dan pembangunan aturcara dalam bahasa pengaturcaraan C. Tujuannya adalah untuk membimbing pengguna dalam menyelesaikan masalah dan menulis aturcara C. Kajian ini menggunakan kaedah penyelidikan campuran, yang mana kaedah kualitatif telah dilaksanakan semasa fasa perancangan keperluan melalui tinjauan literatur yang disokong oleh temubual separa berstruktur, semakan dokumen dan pengesahan kandungan oleh pensyarah pakar pengaturcaraan. Sementara itu, kaedah kuantitatif untuk menilai kebolehgunaan aplikasi dalam kalangan pensyarah yang sama telah dilaksanakan menggunakan soal selidik *System Usability Scale* (SUS) untuk mendapatkan skor purata. Proses pembangunan aplikasi dilaksanakan dengan menggunakan Model *Rapid Application Development* (RAD), yang mana proses prototaip berulang telah menyempurnakan reka bentuk aplikasi, kemudian diikuti dengan pembangunan aplikasi. Kajian ini telah menemui teknik dan reka bentuk yang sesuai untuk persekitaran penyelesaian masalah serta pembangunan aturcara. Dalam persekitaran penyelesaian masalah, konsep pemikiran komputasi (CT) yang disokong oleh Model IPO melalui Arahan dan Inkuiri Saintifik telah digunakan. Manakala persekitaran pembangunan aturcara menampilkan pengaturcaraan berasaskan kerangka melalui Pola Kod. C-SOLVIS telah mencapai skor min SUS 86.07 yang diterjemahkan oleh SUS sebagai gred A, menunjukkan C-SOLVIS adalah sebuah aplikasi yang mempunyai tahap kebolehgunaan yang tinggi untuk digunakan dalam pengajaran dan pembelajaran bagi kursus asas pengaturcaraan. Kesimpulannya, C-SOLVIS mampu memudahkan pengajaran dan pembelajaran pengaturcaraan asas C dengan berkesan. Implikasinya, proses pembangunan C-SOLVIS ini boleh dijadikan garis panduan dalam pembangunan perisian pendidikan khususnya aplikasi pendidikan pengaturcaraan.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ALM	Application Lifecycle Management
CASE	Computer-Aided Software Engineering
CLO	Course Learning Outcome
CORR	Course Outcome Review Report
C-SOLVIS	C Problem-Solving and Visualization
CSUQ	Computer System Usability Questionnaire
CT	Computational Thinking
CVR	Content Validity Ratio
DBMS	Database Management Systems
DEP	Diploma of Electronics (Communication) Engineering
DET	Diploma of Electrical Engineering
DOM	Document Object Model
DTK	Diploma of Electronics (Computer) Engineering
ECMA	European Computer Manufacturers Association
ERD	Entity-Relationship Diagram
GCC	GNU Compiler Collection
GDB	GNU Debugger
GUI	Graphical User Interface
IDE	Integrated Development Environment
IoT	Internet of Things

IPO	Input Process Output
IR4.0	Industrial Revolution 4.0
ISO	International Organization for Standardization
LLDB	Low-Level Debugger
LLVM	Low-Level Virtual Machine
LMS	Learning Management System
MVC	Model-View-Controller
NPM	Node Package Manager
NVM	Node Version Manager
PSAS	Sultan Azlan Shah Polytechnic
PSSUQ	Post-Study System Usability Questionnaire
PT	Python Tutor
PV	Programming Visualization
PVC	PlayVisualizerC
PW	Practical Work
QUIS	Questionnaire for User Interaction Satisfaction
RAD	Rapid Application Development
RO	Research Objective
RQ	Research Question
SBRE	Scenario-Based Requirements Elicitation
SC	SeeC
SDD	Software Design Document
SDK	Software Development Kit
SDLC	Software Development Life Cycle

SOLVEIT	Specification Oriented Language in Visual Environment for Instruction Translation
SPM	Software Process Model
SPSS	Statistical Package for the Social Sciences
SRS	Software Requirement Specifications
STD	Software Test Document
SUS	System Usability Scales
TIOBE	The Importance of Being Earnest
UI	User interface
UML	Unified Modelling Language
VIP	Visual Interpreter
VP	Visual Programming
XML	Extensible Mark-up Language

APPENDIX LIST

CHAPTER 1

INTRODUCTION

1.1 Introduction

In line with the Industrial Revolution 4.0 (IR4.0), computing and software technology is also experiencing rapid development. This revolution has increased the demand for expert programmers. Hence, there is an urge to produce computer science and engineering graduates with good problem-solving and programming skills. These skills are often obtained from programming courses at the higher education level. However, due to its high intrinsic cognitive load, programming has been a challenging lesson to learn among beginners (Cheah, 2020).

Programming needs competency in both problem-solving and program writing (Cheah, 2020; Choi, 2019). However, novice programmers often face difficulties in problem-solving due to their lack of programming experience (Hashim et al., 2017). They are also facing challenges in developing a program syntactically, semantically, and pragmatically which are reflected in their disfluency in writing a program (Henry & Dumas, 2020). It is believed that the implementation of certain problem-solving models and concepts could enhance the problem-solving and programming skills of novices.

The use of a programming Integrated Development Environment (IDE), which is overwhelmed with complex functions is believed to be intimidating for novices (Warner & Guo, 2017). It also does not have facilities to help with problem-solving and programming learning. Several applications were found with various approaches to tackle these issues. However, they are mostly targeting other programming languages where comparatively few are for the C language (Egan & Mcdonald, 2020). Therefore, there is a need for a simple C programming application to introduce novices to an IDE, at the same time guide the novices in problem-solving and programming.

This study aims to develop an introductory IDE, named C Problem-Solving and Visualization (C-SOLVIS), as a tool to enhance the teaching and learning of C programming. For that purpose, C-SOLVIS is designed as a simple introductory IDE to guide problem-solving and facilitate program development using computational thinking (CT) and frame-based programming. The research also evaluates this application's usability specifically on its user interface based on a specific standard.

1.2 Research Background

Since its launch in 2011, the IR4.0 has brought the paradigm shift towards automation and robotics where a majority of the task has been taken over by robots (Kamaruzaman et al., 2019). Therefore, it has caused a change in the demand of the industry required skills. To meet the demand of IR4.0, Chaka (2020) emphasized the importance of problem-solving skills as a generic soft skill that is needed together with the programming skill. Therefore, graduates especially in the engineering and computer science field of study should be prepared with these required skills.

In today's industry, C programming is actively used around the world in solving engineering and scientific problems (Egan & Mcdonald, 2020). Various software such as operating systems, databases, browsers, compilers, and Internet of Things (IoT) applications are developed using the C language. For these reasons, C was once ranked as the most popular programming language among the programming experts in January 2021 (TIOBE Index for January 2021, 2021).

Due to its wide use and popularity, the C language has been an essential and elementary programming language to be taught in universities and educational institutions (Stephen Cass, 2019). It is usually offered for the students in computer science majors and other engineering-related majors during the first year of study. However, several researchers found that this course has recorded high withdrawal and failure rates of 30 – 50%, with a generally low passing rate (Margulieux et al., 2020).

This scenario also occurred in Malaysian higher learning institutions, specifically the Malaysian Polytechnic population. Md Derus (2016) reported that more than 50% of students in Malaysian Polytechnic have achieved below a satisfactory level. In the semester of December 2019, the average percentage of the learning outcome achievement for basic programming knowledge application in one of the Malaysian Polytechnics was only 66%, which is below the target of 80% (Mat Isa & Md Derus, 2017).

An academic report by a course coordinator stated that the low achievement in the programming course was due to students' difficulties in programming which is related to their weaknesses in performing problem-solving (Mat Isa & Md Derus, 2017). This could explain why programming is recognized as one of the great challenges in computing education (Egan & Mcdonald, 2020). Nevertheless, through studies and research, problem-solving can be enhanced by applying a certain problem-solving model to produce a problem-solving algorithm. Thus, by having an algorithm, a program can be written in a specific programming language.

Yet, the disfluency of writing in the programming language has been identified as a significant barrier in programming learning (Edwards et al., 2018). This is supported by Mat Isa & Md Derus (2017), who reported that programming students often face difficulties in learning the language syntax and adapting to the language patterns. They also face difficulties in understanding basic concepts of programming which often lead to misconceptions. As a result, the students often struggle in debugging bundles of syntax errors during programming practical activities.

Practically, students do programming in a program development environment, namely an IDE. However, the IDE is a professional tool that is overwhelmed with tools and facilities. Thus, it imposes an unnecessary extraneous cognitive load for the novices. Hence, Warner & Guo (2017) raised the concern that it can be daunting for novices to understand how to use these tools and facilities while they are already struggling to understand programming mechanisms.

According to the Cognitive Load Theory, the high cognitive load on novices may affect their motivation in learning programming (Abdul Rahman et al., 2018). Therefore, the extraneous cognitive load must be reduced so that the total cognitive load imposed on the students will not exceed their cognitive capacity (Hundhausen et al., 2017). Instead, the germane cognitive load which is useful in the learning process is more needed to help in understanding the course (Sweller et al., 2019).

Although several studies were done to improve programming learning, some other aspects need to be explored as programming remains to be challenging (Shi et al., 2017; Škorić et al., 2016). Therefore, this study focuses on another paradigm which is the integration of problem-solving essences and program development elements. This research contributes to another software design for educational purposes. It is a contribution of the software engineering discipline to improve the teaching methodologies and programming learning materials which is significant in supporting the novices to prepare them for the demand of IR4.0.

1.3 Problem Statement

To score in the programming course, students must be able to solve programming problems and build programs. However, there are three kinds of challenges faced by students when learning to program: problem-solving deficiencies, language difficulties, and ineffective learning tools (Khan et al., 2020), as depicted in Figure 1.1.

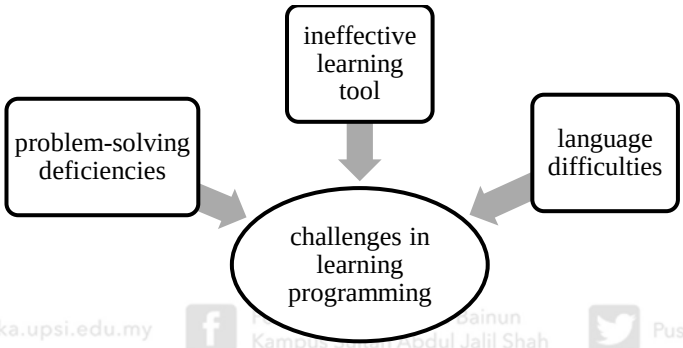

Figure 1.1. Challenges in Learning Programming

The first concern in learning programming is the problem-solving deficiencies among the students. To solve the programming problem, students must be able to understand the problem and formulate solutions into steps that they can translate later into program coding (Nelson et al., 2017). Unfortunately, they tend to have a weak strategic understanding of the problem, thus having difficulties in extracting the needs and requirements of the problem (Kwon, 2017). Moreover, due to their lack of programming experience and lack of systematic problem-solving guidelines, translating the solution into a corresponding computer program could be difficult and challenging (Hashim et al., 2017). Therefore, this research is important to find a suitable technique to help novices in problem-solving and formulate coding solutions.

The second concern is about the students' difficulties in using the language. Their disfluency in C language syntax and unfamiliarity with the language pattern often causes unresolved syntax errors during program writing (Ettles et al., 2018). Moreover, the abstract and dynamic computational execution of the program also imposes great difficulties to understand the program which leads to misconception. Several techniques have been used in programming applications to help in coding and understanding programming (Annamalai & Nur, 2017; Milne & Rowe, 2002). However, few have been applied for C programming (Heinsen Egan & McDonald, 2014). Therefore, this research is carried out to find a suitable technique to help novices in C programming.

The third concern is regarding the use of the programming IDE among the novices, which is an ineffective learning tool. The novices need to get proper training in problem-solving and programming skills. However, most IDE focuses only on program development rather than problem-solving processes although these two skills are needed hand in hand. Moreover, the programming IDE is overwhelmed with programming facilities that are beyond the needs of novices, thus could be intimidating for the novices who are just about to learn to program (Warner & Guo, 2017). Being novices, they should be first introduced to a more suitable IDE with a user-friendly interface before using the professional version. Therefore, this study is an important effort in designing and developing a more suitable educational IDE that caters to both needs of the novices in problem-solving and programming skills.

1.4 Research Objectives

This research aims to develop an introductory IDE that can be used in the teaching and learning of programming courses to enhance both problem-solving skills and programming skills among students and prepare them before using the programming IDE. Therefore, to achieve the purpose, the following are the research objectives (RO):

RO 1: To identify suitable techniques to overcome students' difficulties in learning programming.

RO 2: To design an introductory IDE that integrates guided problem-solving and facilitating program development.

RO 3: To develop an introductory IDE that integrates guided problem-solving and facilitating program development.

RO 4: To evaluate the usability of the introductory IDE to be used in introductory programming courses.

1.5 Research Questions

In achieving the research objectives, several research questions need to be addressed throughout the study. The research questions (RQ) are corresponding to the research objectives, which are listed as the following:

RQ 1: What are the suitable techniques to overcome students' difficulties in learning programming?

RQ 2: How will the introductory IDE be designed to guide problem-solving and facilitate program development?

RQ 3: How will the introductory IDE be developed to integrate guided problem-solving and facilitate program development?

RQ 4: To what extent is the introductory IDE usable in introductory programming courses?

Table 1.1 summarizes the relationship between the research objectives and the research questions.

Table 1.1

Research Objectives and Research Questions

	Research Objectives	Research Question
1	RO 1: To identify suitable techniques to overcome students' difficulties in learning programming.	RQ 1: What are the suitable techniques to overcome students' difficulties in learning programming?
2	RO 2: To design an introductory IDE that integrates guided problem-solving and facilitating program development.	RQ 2: How will the introductory IDE be designed to guide problem-solving and facilitate program development?
3	RO 3: To develop an introductory IDE that integrates guided problem-solving and facilitating program development.	RQ 3: How will the introductory IDE be developed to integrate guided problem-solving and facilitate program development?
4	RO 4: To evaluate the usability of the introductory IDE to be used in introductory programming courses.	RQ 4: To what extent is the introductory IDE usable in introductory programming courses?

1.6 Theoretical and Conceptual Framework of the Research

The research is implemented in two major stages. These two stages are the development stage and the evaluation stage as shown in the theoretical and conceptual framework in Figure 1.2. In the first stage, the development of the application is carried out by using the Rapid Application Development (RAD) Model, which is an incremental software process development model.

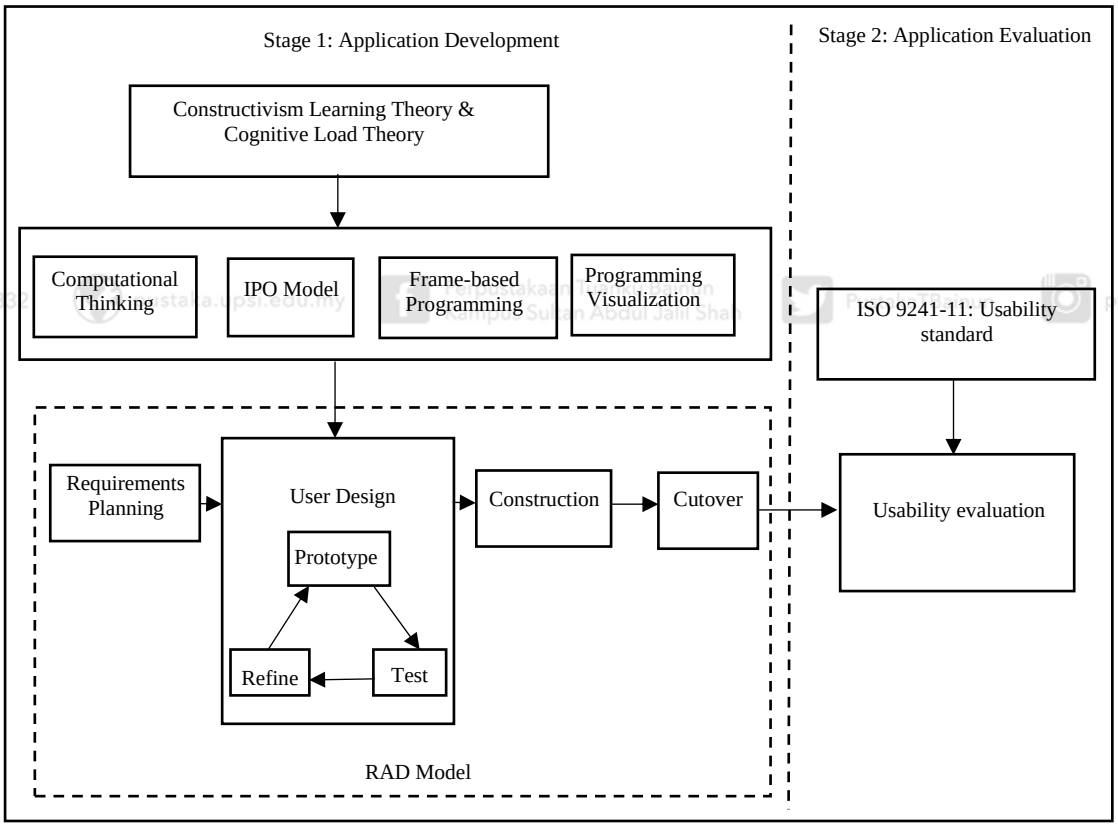

Figure 1.2. Research Theoretical and Conceptual Framework

The RAD model is one of the Software Process Models (SPM) that begins with requirements planning. It is followed by the user design phase where iterative prototype testing and refining process are done until an agreed prototype is produced. Next, it

proceeds with construction activities in the construction phase until the application is completely tested in the cutover phase. Then, in the second stage, the software is evaluated and analyzed by complying with the ISO 9241-11 usability standard.

As the application is intended for the teaching and learning environment, thus it considers some related learning theories. These theories are the Constructivism Learning Theory and Cognitive Load Theory, which are supporting the research to achieve the desired objectives. The Constructivism Learning Theory guides the application design to focus on constructing cognitive experience. Meanwhile, the Cognitive Load Theory encourages germane cognitive load to boost programming learning.

C-SOLVIS applies the Computational Thinking (CT) concept supported by the Input-Proses-Output (IPO) Model to guide the problem-solving formulation (Alshaye et al., 2019). This concept and model help to translate problems into data, mathematical expressions, logical processes, and computational terms in C language programming by identifying the variables in the problem. Meanwhile, the program development is utilizing the frame-based programming concept. In addition, the PV is also applied to enable users among lecturers and students to show and visualize programming execution to understand programming behaviour and structure.

1.7 Research Scope and Limitations

The development of the application is targeting the teaching and learning of the introductory programming course in the C language in higher education in Malaysia. To focus on this scope, this research refers to the Malaysian Polytechnic curriculum which covers the Programming Fundamentals syllabus, focusing on Topic 2: Fundamentals of C Language from the Programming Fundamentals course.

The research focuses on designing and developing an introductory IDE prototype that caters to the problem-solving formulation and programming facilities for basic scientific and engineering problems which involves simple calculation and formulas. Therefore, the application can only be used to solve programming problems that involve basic arithmetic calculations. Although the application design includes the PV interface design, the construction of the prototype focuses on the integration of problem-solving with the program development function which does not cover the construction of PV interfaces.

The application is evaluated on the usability aspect in terms of effectiveness, efficiency and satisfaction by using the ISO 9241-11 standard. For the evaluation purpose, a lab sheet that covers the practical activity involving Operators is used during the application deployment. Although the application is targeting both lecturers and students of the programming course, the evaluation is done only among the lecturers due to several access limitations of the application prototype functions and server.

1.8 Research Significance

This study is significant as it contributes to the body of knowledge in the software engineering field. It also contributes to the programming education field as well as other research communities. The significance of the research is elaborated based on several parties that can benefit from this study as below:

i. **Software engineering body of knowledge**

The success of the research can be used as guidance and reference to software developers and other researchers in software engineering, especially in the field of educational software. This study shows basic procedures and processes in software design and development which contribute to a framework for developing other student-friendly educational software to support educational needs. It contributes as a guide to conducting requirements planning, software design, and construction. Moreover, this study has considered a few learning theories such as the Constructivism Learning Theory and Cognitive Load Theory to help in designing and developing better educational software. Besides, it also contributes to the study of software evaluation that involves the usability evaluation of a software interface design by using specific software evaluation instruments and methods.

ii. **Programming Education**

The introductory IDE which was developed in this study contributes to the production of graduates with a strong foundation of problem-solving and

programming skills as it focuses on accelerating the experience acquisition among the novices based on the Constructivism Learning Theory. A strong foundation in problem-solving and programming skills is essential to becoming an expert programmer. Indirectly, this study contributes to the demand for skilled programmers in IR4.0. Moreover, the study also benefits the lecturers as the features of this application could facilitate the lecturers in guiding problem-solving and teaching the C language syntax among novices as it considers the Cognitive Load Theory in optimizing students' cognitive ability. Therefore, this application also contributes to being an option in the programming instructional tools.

iii. **New knowledge for research communities**

Researchers may gain useful information and knowledge from this study, which allows them to explore and conduct further studies related to programming learning among novices. Since previous studies have focused on the potential of visual programming and programming visualization, this study highlights how problem-solving can be integrated with frame-based programming and programming visualization to enhance the understanding of the subject. The rigorous study which focuses on this integration has also yielded the connection between other knowledge concepts such as CT and IPO Model. It has expanded the ability of CT to be integrated into other models and concepts to produce a new educational approach to facilitating the teaching and learning process.

1.9 Operational Definition

Some terms are being used in this research that needs to be comprehended according to these operational definitions to support the understanding of the research. This section defines important terms as the following:

i. Algorithm

A programming algorithm is the organized, logical sequence of a program, which is represented in the step-by-step computational procedure (Riza et al., 2019). In the research context, an algorithm is the logical flow of the program which is often represented before the program is written and can be obtained by a problem-solving process. It involves calculations or other problem-solving operations done on input variables to get specific outputs.

ii. Cognitive load

A cognitive load is a load that involves a person's intellectual activity such as thinking, reasoning, or remembering (Spieler et al., 2020). In the research context, cognitive load is categorized according to the Cognitive Load Theory as intrinsic cognitive load, germane cognitive load and extraneous cognitive load.

iii. Compiler

A compiler is a special tool that compiles the program written in a high-level language and converts it into a machine-readable object file (Mulla et al., 2016).

In the research context, the compiler is a basic component of the application that combines different object files to create a single executable file so that the program can be run.

iv. Computational Thinking

Computational thinking (CT) is a cognitive skill which is most needed by industries to solve problems with computational solutions, which is also recognized as a 21st-century skill (Lai & Wong, 2022). In this research context, CT has been used in developing instructions and inquiries for solving a programming problem.

v. Debugger

A debugger is used to test and debug a program to locate errors if any so that the error can be fixed (Heinsen Egan & McDonald, 2014). In the research context, the debugger is one of the components needed in the application that can show the position of errors in the program and perform programming visualization.

vi. Extraneous cognitive load

Extraneous cognitive load is the cognitive load that results from an improper instructional design that interferes with the learning process and may reduce instructional effectiveness (Swezller, 1994). In the research context, the extraneous cognitive load refers to any extra burden that should be eliminated to alleviate the cognitive load imposed during the learning.

vii. Frame-based Programming

Frame-based programming is a type of programming that use instruction blocks that contain grouped codes based on their patterns (Sim & Lau, 2018). In the research context, frame-based programming is a technique used to help novices write a C program from the pre-written codes known as Code Patterns with modifiable parameters.

viii. Germane Cognitive Load

Germane cognitive load is a non-intrinsic cognitive load that contributes to learning (Caspersen & Bennedsen, 2007). In the research context, the germane cognitive load is the relevant and useful cognitive load that is needed to enhance the learning process.

ix. Intrinsic Cognitive Load

Intrinsic cognitive load is the fixed cognitive load of a content area which is determined by the relational complexity of the to-be-learned content (Caspersen & Bennedsen, 2007; Swezller, 1994). In the research context, intrinsic cognitive load resembles the difficulty level being imposed by the nature of the subject.

x. Introductory IDE

An introductory IDE is an educational software that is designed to support the learning process (Fiddi, 2015). In the research context, the introductory IDE is similar to a programming IDE but with simpler facilities. It allows the users to write and edit their program to be compiled and executed to get the result. At

the same time, it is supported with educational features with a novice-friendly interface.

xi. Misconceptions

Misconceptions are the misunderstood ideas held by students in programming, which conflict with normally acknowledged programming rules and procedures (Qian & Lehman, 2017). In the research context, misconceptions are the results of poor understanding based on incorrect mental models of the program execution.

xii. Novice programmer

The novice programmer is a computer programmer who is not experienced in programming (Venigalla et al., 2020). In this research context, the novice programmer refers to the students who are learning to program in introductory programming courses. They are unfamiliar with C language programming and inexperienced in performing problem-solving.

xiii. Pragmatic

Pragmatic is the practical knowledge of the environment and language features dictated by practical consequences more than by theory (Deek et al., 1999). In this research context, pragmatic refers to the confidence and knowledge in using the programming environment, which is the IDE. Pragmatic knowledge reflects the higher level of programming skills that are often gained after several periods of experience.

xiv. Problem-solving

Problem-solving is the process of finding the solution for a certain issue or problem. In the research context, problem-solving refers to solving problems using a programming technique. It is the act of defining a problem, translating the problem into computer-compatible form, and determining how to write the solution into a specific programming language to get the problem solved through a computer program.

xv. Program

In the research context, a program is a set of sequential computational instructions that are written in a programming language. It is executed to perform a process that can solve scientific and engineering problems with a computer.

xvi. Program development

Program development is a process of translating a problem solution into a computer program, testing and delivering the solution (Deek et al 2000). In the research context, program development refers to the programming activity to create a computer program that can be used to solve a problem. This term is also used in the C-SOLVIS as one of the main functions, which acts similar to an IDE.

xvii. Program execution

Program execution is the process of running a computer program (Mulla et al., 2016). In the research context, program execution runs the code to tell the

computer to implement instruction step-by-step as written by the programmer which involves variable update according to the instruction.

xviii. Programming

Programming refers to the activity of writing a computer program that needs syntactical, semantical, and pragmatical knowledge of a programming language to solve problems (Scherer et al., 2020). In the research context, programming is related to the activities of writing, compiling, executing, and debugging a program code to solve a specific problem.

xix. Programming Visualization

Programming visualization is the activity to show the internal structure and behaviour of a program that cannot be physically visualized (Shin, 2018). In the research context, visualization is the formation of visual images of the program execution state that shows the result of program execution on variables through a graphical representation.

xx. Prototype

In this research context, a prototype is a preliminary version of the application, from which the application functionalities are developed after the prototype design is agreed upon with all stakeholders (Agarwal et al., 2017). The prototype is subject to the research scope and limitations.

xxi. Semantic

Semantic is the logic interpretation and functional meaning of the written instruction codes in a particular programming language (Deek et al., 1999). In this research context, semantic refers to the programming conceptual knowledge which concerns the logical meaning of the C-language instruction.

xxii. Syntax

The syntax is a set of grammatical rules and approved structural patterns of a specific programming language (Ahadi et al., 2018). In this research context, syntax refers to the set of C-programming grammatical rules and structural patterns that govern the use of valid words and symbols for issuing commands or instructions to make a valid program.

xxiii. Usability

Usability refers to the measure of the ease of use for the user in using the application (Komiyama et al., 2020). In the research context, usability refers to the measure of effectiveness, efficiency and satisfaction of the application's user interface which refers to the ISO 9241-11 standard (Komiyama et al., 2020).

xxiv. Variable

From a program perspective, a variable is an element in a program that holds data with a descriptive name, which is allocated to a specific memory location

(Kohn, 2017). In the research context, a variable represents any data or information that will be operated by programming operations and processes.

1.10 Summary

The research is about overcoming the challenges in programming teaching and learning which intends to help lecturers in teaching programming with better application and help the students who are novice programmers in learning programming especially in problem-solving and developing C programs. Therefore, the C-SOLVIS is proposed to be developed as an introductory IDE for use in the Programming Fundamentals course. C-SOLVIS which is developed based on a software development model integrates problem-solving essence into a program development environment. Supported with established models, concepts and learning theories, it aims to build a strong programming foundation to prepare the students for the next challenge in the era of IR4.0.

The next chapter reviews the related studies on C programming learning that discovers the difficulties of students in learning programming. Besides, several related learning theories, models and studies in problem-solving and programming are also reviewed that initiates the introductory IDE development ideas. It also discusses the introductory IDE development criteria, software process model for software development and several standards in the evaluation of software usability.