

**AGENT BASED SYNTAX ERROR DETECTION
FOR C PROGRAMMING LANGUAGE**



SYRA G MOKUNJIL

UMS
UNIVERSITI MALAYSIA SABAH

**SCHOOL OF ENGINEERING AND
INFORMATION TECHNOLOGY
UNIVERSITI MALAYSIA SABAH
2013**

**AGENT BASED SYNTAX ERROR DETECTION
FOR C PROGRAMMING LANGUAGE**

SYRA G MOKUNJIL



UIMS

**THESIS SUBMITTED IN FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE**

**SCHOOL OF ENGINEERING AND
INFORMATION TECHNOLOGY
UNIVERSITI MALAYSIA SABAH
2013**

UNIVERSITI MALAYSIA SABAH

BORANG PENGESAHAN TESIS

JUDUL : _____

IJAZAH : _____

SAYA : _____ SESI PENGAJIAN : _____

(HURUF BESAR)

Mengaku membenarkan tesis *(LPSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Universiti Malaysia Sabah dengan syarat-syarat kegunaan seperti berikut:-

1. Tesis adalah hak milik Universiti Malaysia Sabah.
2. Perpustakaan Universiti Malaysia Sabah dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (/)

SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di AKTA RAHSIA RASMI 1972)

TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD

Disahkan oleh:

(TANDATANGAN PENULIS)_____
(TANDATANGAN PUSTAKAWAN)

Alamat Tetap: _____

TARIKH: _____

(NAMA PENYELIA)

TARIKH: _____

Catatan:

*Potong yang tidak berkenaan.

*Jika tesis ini SULIT dan TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis ini perlu dikelaskan sebagai SULIT dan TERHAD.

*Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana Secara Penyelidikan atau disertai bagi pengajian secara kerja kursus dan Laporan Projek Sarjana Muda (LPSM).

DECLARATION

I hereby declare that the material in this thesis is my own except for the quotations, excerpts, equations, summaries and references which have been duly acknowledged.

15th May 2013

Syra G Mokunjl
PS05-008-002



UMS
UNIVERSITI MALAYSIA SABAH

CERTIFICATION

NAME : **SYRA G MOKUNJIL**
MATRIC NO : **PS05-008-002**
TITLE : **AGENT BASED SYNTAX ERROR DETECTION FOR
C PROGRAMMING**
DEGREE : **MASTER OF SCIENCE**
VIVA DATE : **31st JANUARY 2012**

DECLARED BY

1. SUPERVISOR

Associate Professor Dr. Patricia Anthony

Signature



UMS
UNIVERSITI MALAYSIA SABAH

ACKNOWLEDGEMENT

The author wishes to extend her gratefulness to...

Associate Professor Dr. Patricia Anthony, for her guidance, motivation, advice, patience and support throughout this research. Her continuous encouragement provided the necessary impetus to complete the research and publish this thesis.

Her special thanks go to the staffs of Media and Educational Technology Unit, University Malaysia Sabah. Their assistance in networking facility is instrumental in materializing OSAAS. Her acknowledgement also goes to UMS library, Postgraduate Center and Sekolah Kejuruteraan dan Teknologi Maklumat, Universiti Malaysia Sabah staff.

Special thanks especially go to Ms. Adriana Boitol of Jabatan Perkhidmatan Komputer Negeri, Kota Kinabalu for her unending assistance, advice and help throughout the early OSAAS development.

Greatest appreciation also goes to her family especially her beloved parents, Noneh @ Margaret Datuk Sibidol and George Anthony Mokunjil for their unconditional love and support throughout the years. To her loving husband, Anselmius @ Robert Kimon who has been her pillar of inspiration and strength. To all her lovely and wonderful children: Odysseus Yabong, Donovan Sibidol, Crescynthia Kinsia and Grace Jitulip who makes her smile everyday.

Last but not least, the author extends her sincere thanks to all her family and friends as well as those who have contributed directly and indirectly towards the success of her project. Thank You.

Thank you God for helping me during the toughest time of this project. Amen.

Syra G Mokunjil
15th May 2013

ABSTRAK

Ramai pelajar universiti mendapati tugas menyelesaikan kesilapan sintak hasil daripada pengkompilan dalam pengaturcaraan adalah mencabar kerana mereka tidak tahu bagaimana dan dari mana hendak bermula. Walaupun terdapat banyak maklumat yang boleh diperolehi samada dalam bentuk cetakan atau pun daripada internet tetapi jumlah maklumat yang terlalu banyak pada masa kini hanya akan membuat pelajar lebih keliru apabila mencari maklumat yang menepati kehendak semasa. Ini menyebabkan pelajar akan mengambil masa yang lama untuk menyiapkan tugas mereka. Tugas yang dihantar secara manual ke fakulti pula berkemungkinan boleh hilang dan oleh itu, pelajar perlu membuat semula tugas mereka. Pensyarah pula perlu menyediakan soalan-soalan tugas, mengingatkan pelajar tentang tarikh akhir penghantaran tugas, menjawab soalan-soalan pelajar dan sebagainya. Tugas-tugas seperti ini banyak memakan masa pensyarah. Oleh itu, sebuah system aplikasi dalam kerja penyelidikan *Agent Based Syntax Error Detection for C Programming* iaitu *Online Student Assignment Assistance System (OSAAS)* telah dibangunkan untuk mengautomatiskan sebahagian daripada tugas-tugas yang dinyatakan tadi. Sistem ini berupaya untuk mengkompil bahasa pengaturcaraan melalui laman web OSAAS dan sekiranya kesilapan sintak dikesan selepas fail dikompil, sistem ini akan memberi penerangan mengapa kesilapan sintak tersebut berlaku. OSAAS kemudian akan memberi cadangan bagaimana kesilapan sintak itu boleh diselesaikan. Teknik agen adalah diaplikasikan didalam sistem ini untuk mengesan kesilapan sintak program dan juga untuk menjana bantuan yang berkaitan dengan kesilapan yang dikesan tadi. Sistem ini juga mempunyai fasiliti untuk mengingatkan pelajar tentang tarikh akhir tugas perlu dihantar, bantuan pengetahuan secara 'online' serta fasiliti untuk pensyarah memberi komen ke atas tugas serta memasukkan markah pelajar. Oleh itu, adalah diharapkan agar OSAAS dapat memberikan motivasi untuk pelajar mempelajari bahasa pengaturcaraan dengan cara yang lebih mudah serta menarik dan juga menjadi dorongan untuk menyiapkan tugas mereka dengan lebih awal.

ABSTRACT

AGENT BASED SYNTAX ERROR DETECTION FOR C PROGRAMMING

A lot of university students find debugging syntax errors overwhelming especially if they are novice to the world of programming language. This is because they do not know how or where to begin. Although there are enormous printed and online materials available but the information 'overloading' may only result in more confusion for the students in seeking for the right information. Handing the assignment manually to the faculty may pose the risk of the assignment being lost and thus, student has to redo and resubmit the assignment. On the other hand, lecturer has to spend time on mundane and repetitive task of preparing assignment questions, reminding students of submission deadline, answering queries etc. Thus, an online web-based system in Agent Based Syntax Error Detection for C Programming named Online Student Assignment Assistance System (OSAAS) is developed to automate some of the human repetitive and mundane task mentioned earlier. It applies the intelligent agent technique whereby it is able to compile the programming file through the website and directs student to the related help file upon detection of syntax error. Submission deadline reminders, simple online help search, marking and grading facilities are some of the useful features of OSAAS. Thus, it is hoped that OSAAS will encourage early assignments submission and provide greater learning satisfaction.



UMS
UNIVERSITI MALAYSIA SABAH

TABLE OF CONTENTS

	Page
TITLE	
DECLARATION	i
CERTIFICATION	ii
ACKNOWLEDGEMENT	iii
ABSTRAK	iv
ABSTRACT	v
TABLES OF CONTENTS	vi
LIST OF TABLES	x
LIST OF ILLUSTRATIONS	xi
LIST OF ABBREVIATION	xiv
CHAPTER 1 : INTRODUCTION	
1.1 Introduction	1
1.2 Conventional Assessment Flow at SEIT	2
1.3 Existing C Integrated Development Environment Setbacks	4
1.4 Rationale and the Need for an Agent-based Syntax Error Detection	8
1.5 Objectives of the Research	10
1.6 Scope of Research	11
1.7 Organisation of Thesis	12
CHAPTER 2 : LITERATURE REVIEW	
2.1 Chapter Overview	14
2.2 Introduction	14
2.3 E-Learning	15
2.3.1 E-Learning Definition	16
2.4 Agent	18
2.4.1 Agent Definition	19
2.5 Applications on Agent and E-Learning	21
2.5.1 C Programming on UNIX	21

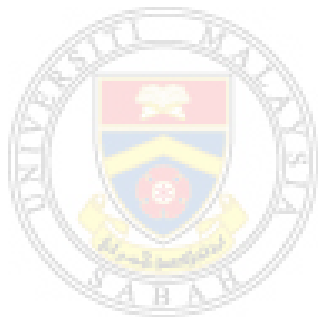
2.5.2	eXtreme e-Learning eXperience (xLx)	24
2.5.3	Knowbots	28
2.5.4	Online Assignment Submission and Management (OASM)	34
2.5.5	Submit! Automatic Program Critiquing	36
2.5.6	CourseMarker CBA System	37
2.5.7	Java Intelligent Tutoring System (JITS)	42
2.6	Review of Literature on Agent-Based e-Learning Systems	45

CHAPTER 3 : OSAAS ARCHITECTURE AND FRAMEWORK

3.1	Chapter Overview	47
3.2	Introduction	47
3.3	OSAAS e-Learning Environment	48
3.4	OSAAS Design	50
3.4.1	OSAAS Goal Overview	51
3.4.2	OSAAS System Overview	53
3.4.3	Assignment Management Agent Overview	55
3.4.4	Assignment Checking Agent Overview	57
3.4.5	Assignment Guidance Agent Overview	58
3.5	OSAAS Submit Assignment Module	59
3.6	Assignment Checking Agent Implementation	62
3.6.1	Missing 'Left' Parenthesis	62
3.6.2	Missing Semicolon	65
3.7	Assignment Checking Agent Algorithm	67
3.8	Assignment Guidance Agent Algorithm	70
3.9	Online C Help	79
3.10	Assignment Reminder	80
3.11	Assignment Result	81
3.12	Lecturer Module	81
3.13	Assignment Submission Module	82
3.14	Assignment Grading	83
3.15	OSAAS Development Software	85
3.16	Summary	85

CHAPTER 4 :		EXPERIMENTAL EVALUATION AND ANALYSIS	
4.1	Chapter Overview		86
4.2	Introduction		86
4.3	Hypotheses		87
4.4	Accuracy of Related Help		87
	4.4.1	Without Error Source Codes and Compiles Successfully	88
	4.4.2	Incorrect Source Codes and Redirection To Related Help	89
	4.4.3	Source Codes Containing 'Warning' Statement and Still Compiles	92
4.5	Survey Questionnaires		94
4.6	Evaluation Statistics on Survey Questionnaires		94
4.7	Survey Questionnaire Analysis		96
	4.7.1	OSAAS as a Motivator	97
	4.7.2	OSAAS C Programming Syntax Error Detection	98
	4.7.3	OSAAS Related Help	99
	4.7.4	OSAAS Online Help	100
	4.7.5	Reduce Time to Learn Programming	101
	4.7.6	OSAAS as an Assignment Deadline Reminder	102
	4.7.7	OSAAS Overall Content	102
4.8	Assignment Management Analysis by Lecturer		103
4.9	Borland C++ versus OSAAS		104
4.10	C Programming on UNIX versus OSAAS		105
4.8	Conclusion		105
CHAPTER 5 :		EXPERIMENTAL EVALUATION AND CONCLUSION	
5.1	Project Summary		107
5.2	Novel Contribution		110
5.3	Future Work		110

REFERENCES	113
APPENDIX A : FIRST SURVEY QUESTIONNAIRE FOR - STUDENT'S PERSPECTIVE	118
APPENDIX B : SECOND SURVEY QUESTIONNAIRE FORM – STUDENT'S PERSPECTIVE	120
APPENDIX C : SURVEY QUESTIONNAIRE FORM – LECTURER'S PERSPECTIVE	122
APPENDIX D : LIST OF PUBLICATIONS	123



UMS
UNIVERSITI MALAYSIA SABAH

LIST OF TABLE

		Page
Table 2.1	Ceilidh's Basic Tools for the Assessment of Programming Related Course	39
Table 2.2	User's Responsibility at Different Levels of System Use – CourseMarker vs Ceilidh	41



UMS
UNIVERSITI MALAYSIA SABAH

LIST OF ILLUSTRATIONS

	Page	
Figure 1.1	Unclear Meaning of Syntax Errors	5
Figure 1.2	'Expression Syntax Errors' Error Message	6
Figure 1.3	'While Statement Missing' Error Message	6
Figure 1.4	Approximation of Line Number for Statement Which Contains Syntax Error	7
Figure 2.1	E-Learning System and Involved Roles	18
Figure 2.2	Agent Architecture	20
Figure 2.3	Concept of Support System for C Programming in UNIX	22
Figure 2.4	The Process of Debug Agent	23
Figure 2.5	Integrative Approach of the xLx-Java Testing Module	26
Figure 2.6	Annotation Window from the Student's Point of View	27
Figure 2.7	General Architecture of Knowbot	30
Figure 2.8	Internal Architecture of Knowbot	31
Figure 2.9	A Sample Screen Shot of Resulting Page (or Report) From a KnowBot for Assignment #1-2 in the ALN Workshop	32
Figure 2.10	A Sample Screen Shot of Detailed Analysis From a KnowBot for Assignment #1-2 in the ALN Workshop	33
Figure 2.11	Student's View of Ceilidh Text Menu Interface	38
Figure 2.12	Exercise Result for a Student	40
Figure 2.13	CourseMarker Graphic User Interface	42
Figure 2.14	JITS User Interface	44
Figure 3.1	OSAAS E-Learning Environment	49
Figure 3.2	Goal Overview for Assignment Management	52
Figure 3.3	Goal Overview for Assignment Checking	52
Figure 3.4	Goal Overview for Assignment Guidance	53
Figure 3.5	OSAAS System Overview	54
Figure 3.6	Assignment Management Agent Overview Diagram	56
Figure 3.7	Assignment Checking Agent Overview Diagram	57
Figure 3.8	Assignment Guidance Agent Overview Diagram	58
Figure 3.9	OSAAS Upload Assignment Menu	60
Figure 3.10	Overview of C Source Code Checking by the Compiler	61
Figure 3.11	OSAAS Uploaded Assignment Menu	62

Figure 3.12	C Source Codes Containing Missing Left Parenthesis Syntax Error	63
Figure 3.13	OSAAS Compiler Output for Missing Left Parenthesis	64
Figure 3.14	Related Help for Missing Left Parenthesis Syntax Error	65
Figure 3.15	C Source Codes Containing Missing Semicolon Syntax Error	66
Figure 3.16	Compiler Output for Missing Semicolon	66
Figure 3.17	Related Help for Missing Semicolon	67
Figure 3.18	Code to Invoke the C Compiler at the Command Line	68
Figure 3.19	Borland C++ 4.5 Compiler Command via DOS Command Line	68
Figure 3.20	Assignment Checking Adapted from Schwieran (2006)	69
Figure 3.21	Actual Line and Statement Showing C Statement Syntax Error	71
Figure 3.22	VB.Net Coding to Get Line Number Which Contains Syntax Error	71
Figure 3.23	VB.Net Coding to Read Line Number Which Contains Comment or Blank Statement	73
Figure 3.24	Line Number to Show C Statement Code Which Contains Syntax Error	74
Figure 3.25	VB.Net Coding To Display the Syntax Error Statement	74
Figure 3.26	Related Online Help Upon Detection of Syntax Error	75
Figure 3.27	Example of Syntax Error in the Source Code	75
Figure 3.28	Example on 'How' to Solve the Missing Parenthesis Error	76
Figure 3.29	VB.Net Coding to Check for Parenthesis In C Source Codes	77
Figure 3.30	VB.Net Coding to Check for Semicolon In C Source Codes	78
Figure 3.31	Example on 'How' to Solve the Missing Semicolon	78
Figure 3.32	OSAAS C Online Help	79
Figure 3.33	OSAAS Assignment Reminder	80
Figure 3.34	OSAAS Individual Student Result	81
Figure 3.35	OSAAS Assignment Submission	83
Figure 3.36	OSAAS Assignment Marking Menu	83
Figure 3.37	Student's C Source Codes Example	84
Figure 4.1	Compiler Result for Source Code Without Syntax Error	88
Figure 4.2	Display in Student Module to Indicate Assignment Successfully Compiled	89
Figure 4.3	Code that Contains Syntax Errors and Linked to Related Help	90

Figure 4.4	Related Help Provided by the Assignment Guidance Agent	91
Figure 4.5	Code Contains Syntax Error But No Related Help	92
Figure 4.6	Warning Statements Generated by the C Compiler	93
Figure 4.7	Related Help for 'Warning' Statement	93
Figure 4.8	Summary Result of First Survey Questionnaires	96
Figure 4.9	Summary Result of Second Survey Questionnaires	97



UMS
UNIVERSITI MALAYSIA SABAH

ABBREVIATION

OSAAS	Online Student Assignment Assistance System
CBA	Computer Based Assessment
CM	CourseMarker
CPU	Central Processing Unit
F2F	Face To Face
GUI	Graphic User Interface
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
ITS	Intelligent Tutoring System
JECA	Java Error Correction Algorithm
JITS	Java Intelligent Tutoring System
LAN	Local Area Network
LCMS	Learning Content Management System
LMS	Learning Management System
OASM	Online Assignment Submission and Management
OL	Online Learning
PLUMS	Pusat Pendidikan Luar Kampus
RTF	Rich Text Document
SEIT	School of Engineering & Information Technology
UMS	University Malaysia Sabah
URL	Uniform Resource Locator
WWW	World Wide Web
xLx	xTreme Learning eXperience

CHAPTER 1

INTRODUCTION

1.1 Introduction

This research work studies the effectiveness of adopting agent technology technique on syntax error detection for C programming language assignment done by computing undergraduates at School of Engineering and Information Technology (SEIT), University Malaysia Sabah. The syntax error detection mechanism forms an integral part of an online web-based assignment submission system. This e-learning system is developed to assist students in their programming assignments starting from initial development to final submission of the students' assignments. The proposed system is called Online Student Assignment Assistance System (OSAAS). OSAAS is a web-based system that could perform some of the human repetitive tasks such as monitoring and management of course activities. Initial study looks into the setbacks and challenges faced by the students in completing their C programming assignments. Improving the current practice of learning C programming can be perceived in many different ways, such as faster assignment submission, reducing wasted time of relearning and improving learner satisfaction.

Students studying programming course learns to code best by practical application of the programming knowledge in order to acquire the skills. Programming cannot be learnt without doing a lot of practice (Ala-Mutka, 2005). Thus, to realize this, the university gives programming assignments to students for which students will complete and submit for marking. Assignment forms an integral part of the continuous assessments and this method of assessment normally contributes some 20 to 30 percent of the overall mark of a given course. The other forms of assessment may be periodic test and final examination.

However, learning a new programming language is very challenging let alone developing a simple system. Student has much to learn in the first programming class such as familiarizing with the Integrated Development Environment (IDE) of new software, application of the language syntax, logical sequence of coding steps, making sense of compiler error messages, appropriate formatting, modularized functionality and understanding complex programming concepts. It is natural to feel less efficient and even lost in the beginning of learning a new programming language as students are in a new and unknown environment and a lot of time will go into looking for information. A big problem of learning a new programming language is that the student will have immense difficulty in understanding any source codes or example codes. It becomes a nightmare when a simple program created generates countless syntax errors. Teaching one self to overcome these challenges involves more than just reading a book and playing around. Sometimes it involves taking time to investigate the parts such as compiler messages that do not immediately make sense. One or two sources of learning are insufficient to develop a basic program. Most students especially novice programmers tend to immerse themselves into loads of online or printed materials to look for the right information to solve these syntax and logic errors which may only result in further confusion.

Thus, as mentioned earlier, OSAAS which forms as part of agent-based syntax error detection for C programming research work is developed to address these issues and challenges. Emphasis is focused on providing guidance to students on solving syntax errors problems generated by the compiler.

1.2 Conventional Assessment Flow at SEIT

One of the assessment methods used to assess student's aptitude in programming courses in SEIT is by assignment. The conventional practice begins with the lecturer preparing the programming assignment's questions at the beginning of the semester. This printed paper-based assignments tasks are normally distributed in the lecture class during the second week of the semester. Students are required to complete and submit their completed assignment by the 8th and 13th weeks of

lectures. Thus, students will only have approximately eight weeks to study the new programming subject and submit their completed assignments by the due date.

Normally, a programming assignment task requires students to create a small application to assess their understanding and application of the subject matter. Students will need to write the codes, compile and run their completed system before finalizing it for submission. These source codes are compiled and executed manually through the IDE for C. Students at SEIT can have the choice to develop their C programming assignments using any IDE for C as long as the platform used is compliant with ANSI C standard. Although, some IDEs such as Eclipse and MS Visual Studio.net offers specific help on syntax error messages, most students at SEIT uses other various IDEs to develop their assignment. Thus, this research work applies the agent technique and sees how it will be applicable in this context.

To many new students, some syntax errors or warnings generated by the C compiler of the IDEs they use to write their source code may not be very understandable. Novice programmer finds it not only difficult but confusing to look for solutions to the syntax errors. Thus, students need to do a lot of research and reading in order for them to write good programming codes.

Once their program files are free of errors, students will then save these files in the thumb drive or compact disks and print out the hardcopy of the programming documentations such as source codes, algorithm and user manuals. Completed assignments which comprises of softcopy and hardcopy of the system are then manually handed to the lecturer or faculty before or by the due date.

Lecturer will start grading each of the assignments by first going through the source codes and documentation files read from the compact disk or thumb drive. Every source code will have to be compiled, test run and executed. Apart from that, lecturer will also assess and mark the documentations attached. Documentations and hardcopy of the source codes are annotated with written comments explaining the strengths and weaknesses of the work submitted, as well

as some suggestions for improvement. Marks are handwritten in a small piece of paper comprising a standard grading format which will be attached to the marked assignments. The final mark is computed by summing up the marks for the softcopy and the hardcopy. These assignments will then be placed in a box for students to collect.

On top of these routine tasks, lecturer also has to constantly remind students of the assignment deadline. Some of the methods of reminding students for assignments which are not yet submitted to the lecturer are by reminding them during lecture as well as listing out the students' names, printing and posting it on the announcement board.

1.3 Existing C Integrated Development Environment Setbacks

The original motivation for the development of OSAAS was driven by a number of setbacks mentioned in section 1.2 above. The problems stated below are the foremost reason OSAAS is developed to address:

a. **Difficulty in Understanding the Meaning of Syntax Errors**

One of the uncertainties that a beginner student may face is the difficulty in interpreting the meaning of the syntax errors provided by some compiler such as provided by Borland C++. An example of this is shown in Figure 1.1. The first line of error which is "Declaration Syntax Error in function main" states that line 15 of the coding contains incorrect syntax declaration, however, it does not explain the kind of declaration that contains error nor does it give any detailed guidance to solving the error. The highlighted line shows the possible location of an error which has occurred in the source code, however, it does not show exactly where in the source code the error has occurred. In this example, the error does not state that the missing equal symbol "=" is the possible cause of the error. Thus, the actual cause of error has to be figured out.

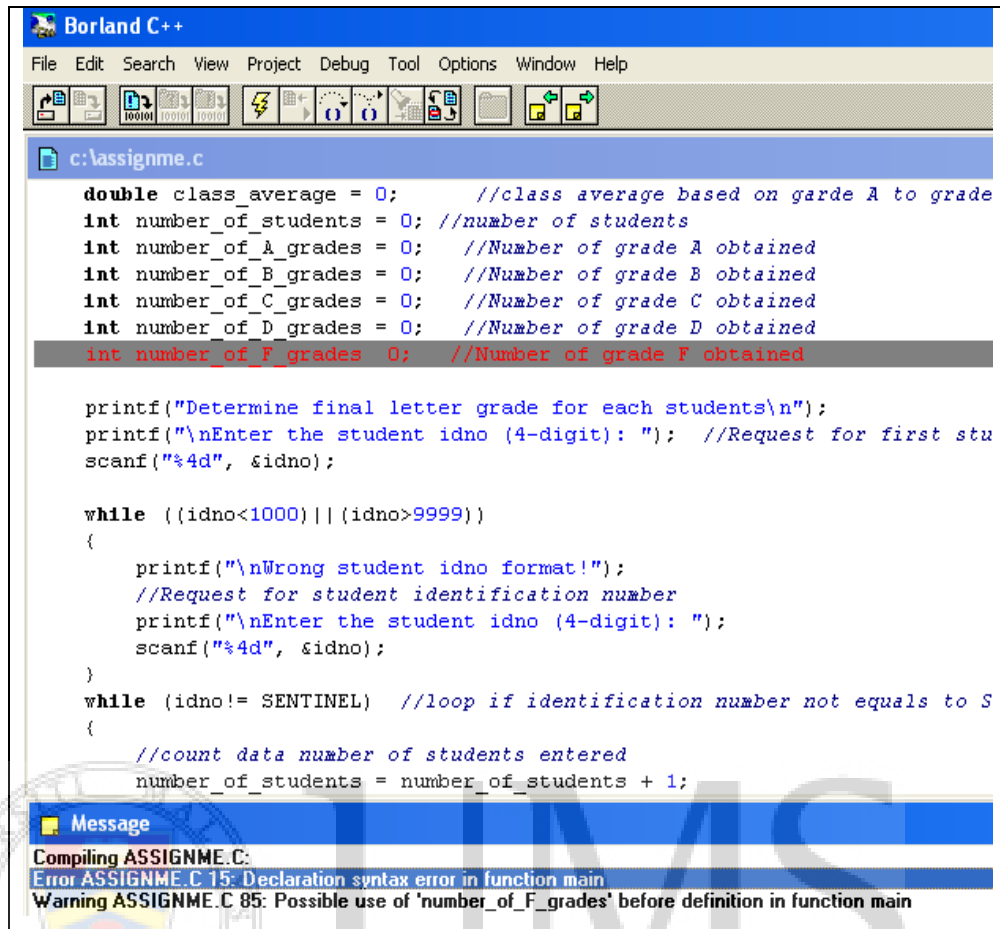


Figure 1.1 : Unclear Meaning of Syntax Errors

b. Different Errors Messages for Comparable Syntax Errors

Compiler sometimes generates different error messages for comparable syntax errors. Figure 1.2 and Figure 1.3 highlighted the same 'While' statement code in line 21. While Figure 1.3 shows "While Statement missing) in function main" error, Figure 1.2 shows "Expression syntax in function main" error. However, in both Figures, both 'While' statements are actually missing one parenthesis with Figure 1.2 missing the left parenthesis while in Figure 1.3, the right parenthesis is missing. Thus, while both statements are missing one of the other side of the parenthesis, but the compiler generated different kind of syntax errors. These kinds of syntax errors generated by the compiler might cause some confusion to students who are new to programming.

```

Borland C++
File Edit Search View Project Debug Tool Options Window Help
c:\assignme.c
double class_average = 0; //class average based on garde A to grade
int number_of_students = 0; //number of students
int number_of_A_grades = 0; //Number of grade A obtained
int number_of_B_grades = 0; //Number of grade B obtained
int number_of_C_grades = 0; //Number of grade C obtained
int number_of_D_grades = 0; //Number of grade D obtained
int number_of_F_grades = 0; //Number of grade F obtained

printf("Determine final letter grade for each students\n");
printf("\nEnter the student idno (4-digit): "); //Request for first stu
scanf("%4d", &idno);

while (idno<1000)|| (idno>9999)
{
    printf("\nWrong student idno format!");
    //Request for student identification number
    printf("\nEnter the student idno (4-digit): ");
    scanf("%4d", &idno);
}
while (idno!= SENTINEL) //loop if identification number not equals to S;
{
    //count data number of students entered
    number_of_students = number_of_students + 1;
}

```

Message
Compiling ASSIGNME.C:
Error ASSIGNME.C 21: Expression syntax in function main
Warning ASSIGNME.C 115: 'grade' is declared but never used in function main
Warning ASSIGNME.C 115: 'semester_average' is assigned a value that is never used in function main
Warning ASSIGNME.C 115: 'final_test' is declared but never used in function main
Warning ASSIGNME.C 115: 'test2' is declared but never used in function main
Warning ASSIGNME.C 115: 'test1' is declared but never used in function main

Figure 1.2 : 'Expression Syntax Errors' Error Message

```

Borland C++
File Edit Search View Project Debug Tool Options Window Help
c:\assignme.c
double class_average = 0; //class average based on garde A to grade I
int number_of_students = 0; //number of students
int number_of_A_grades = 0; //Number of grade A obtained
int number_of_B_grades = 0; //Number of grade B obtained
int number_of_C_grades = 0; //Number of grade C obtained
int number_of_D_grades = 0; //Number of grade D obtained
int number_of_F_grades = 0; //Number of grade F obtained

printf("Determine final letter grade for each students\n");
printf("\nEnter the student idno (4-digit): "); //Request for first stude
scanf("%4d", &idno);

while (idno<1000)|| (idno>9999)
{
    printf("\nWrong student idno format!");
    //Request for student identification number
    printf("\nEnter the student idno (4-digit): ");
    scanf("%4d", &idno);
}
while (idno!= SENTINEL) //loop if identification number not equals to SEN
{
    //count data number of students entered
    number_of_students = number_of_students + 1;
}

```

Message
Compiling ASSIGNME.C:
Error ASSIGNME.C 21: While statement missing } in function main
Warning ASSIGNME.C 115: 'grade' is declared but never used in function main
Warning ASSIGNME.C 115: 'semester_average' is assigned a value that is never used in function main
Warning ASSIGNME.C 115: 'final_test' is declared but never used in function main
Warning ASSIGNME.C 115: 'test2' is declared but never used in function main
Warning ASSIGNME.C 115: 'test1' is declared but never used in function main

Figure 1.3 : 'While Statement Missing' Error Message

c. Inaccurate Line Number

Compiler may also display inaccurate line number of the statement that contains syntax error. The actual line number that contains error may not be the one highlighted by the IDE, but the line that precedes the highlighted statement.

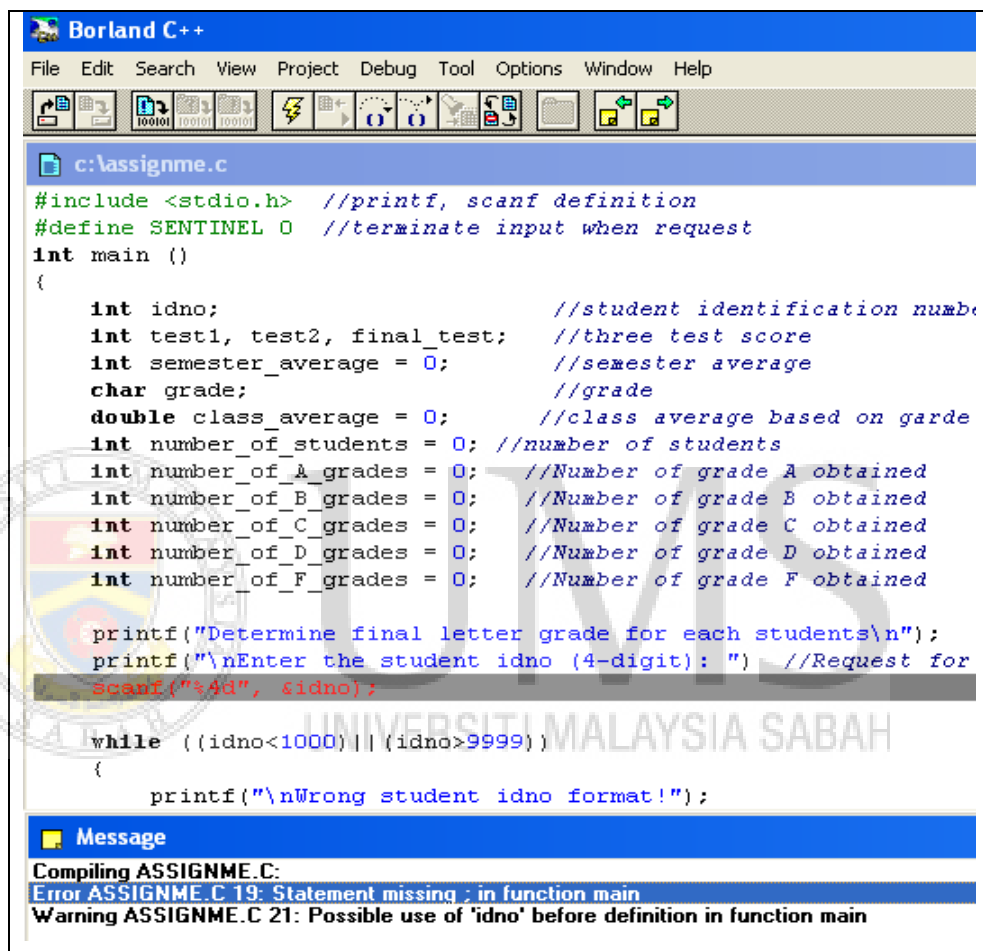


Figure 1.4 : Approximation of Line Number for Statement Which Contains Syntax Error

Figure 1.4 shows an example of syntax error message displaying approximate location of the statement that contains syntax error. The figure shows that line number 19 is highlighted and contains the said "Statement missing ; in function main" error. However, the actual error which is the missing ` ; ` is actually in line 18. Line 19 is actually a correct statement but