6th International Conference on Research Developments in Applied Science, Engineering & Management
Mahratta Chamber of Commerce, Industries and Agriculture, Pune (India)
23rd March 2019 www.conferenceworld.in ISBN : 978-93-87793-78-1

Computer based Programming Assistant Tool for C-programming Beginners

Prachi Gupta, Surendra Gupta
Department of Computer Engineering, Shri G.S. Institute of Tech. and Science, 23, Sir Visvesvaraya Marg, Indore (MP), India
Department of Computer Engineering Shri G.S. Institute of Tech. and Science 23, Sir Visvesvaraya Marg, Indore (MP), India

Abstract— Many students who are starting their studies in Engineering don't have any knowledge about programming because they haven't studied programming earlier. Thus when students start writing programs in C language, they don't understand the grammar of C programming language, so they make lot of syntax errors. Programming Assistant tool assist first year student with the programming problems. In this research work, Focus is only on auto correction of Syntax errors. This research work focus on auto correction of syntax errors using computer based programming assistant tool. So here in this research work, Some of the problem has been tried to solve which has been faced by engineering students in the initial stage of programming. So if Compile time errors will get corrected then student can understand logic rather than being stuck on syntax errors. A compile time error is consist of problems such as a syntax error or missing file reference that prevents the program from successful compilation. In this research work, most commonly found compile time errors made by students get corrected. Implementation of programming assistant tool will help the students and teachers to save time during programming labs.

Keywords - Compile time errors, Programming assistant tool.

I. INTRODUCTION

For first year students, It is very essential to understand the importance of programming languages. Understanding of programming language is very important for engineering students. It is very important to make students realize what programming problems they are going to face in their initial stage. They need to understand the problem because in the future they are only ones who will provide solution to those problems.

The key objective of designing programming assistant tool is to make student aware of programming logics rather than they will just keep correcting the syntax errors. Syntax errors are errors which occurred during writing of the syntax of program. A well written code is passed to the compiler, which converts code from one language(High Level lan-guage) to another(Low level language).

Compiler can only convert a code of one language to another only when syntax of the code is written correctly otherwise it will create some compile time errors.

Existing methods provide auto completion for the key-words but in auto completion user have to select keywords from the drop down menu. Existing methods provides lots of facilities similar to auto completion. Although auto com-pletion has been already done,some students are still unable to select the keywords from the drop down menu and they do make mistakes with special character as well .

ISBN : 978-93-87793-78-1

Therefore, in this paper, Computer Based programming assistant tool has been designed. As most of the students coming from the non technical background so they aren't able to write code perfectly or we can say that without compile time errors.This method will be usefull for the student to learn programming logic rather than just being stuck on the Syntax errors. If syntax errors get corrected automatically then students can develop their logical thinking and there interest towards programming will keep growing.

This paper is sorted out as pursues. In segment II a diagram of work officially done here with their points of interest and inconveniences is talked about. In the segment,

III the proposed methodology is exhibited. Segment IV introduces the subtleties of test investigation of the outcome. Conclusion and future work is examined in area V.

II. RELATED WORK

Substantial work has been carried out by earlier researchers in the field of programming assistant tools. Some of the most relevant work is discussed in this section.

For decades, researchers and instructors have been trying to enhance or improve the learning process of students. In this process, it is important to know whether students have misconceptions in their conceptual understanding. The study of these elements is becoming a relevant research area in science and engineering education[5].So the researcher tried to enhance the learning process and tried to assure that students doesn't have any kind of misconception.

As we all know in their precollege days,students only tends to learn things which are available in books but they doesn't have any practical knowledge.Numerous efforts has been made to extend awareness, interest, and participation in scientific and technological fields at the precollege level. Studies have shown these students are at a crucial age wherever exposure to engineering and different connected fields like science, arithmetic, and technology greatly impact their career goals[4].

To increase interest of students towards technology, colleges has made enormous effort. Those efforts can be organizing seminars by experts, organizing workshops by industry officials, rewarding students who are making serious effort towards technological knowledge etc.

1. Errors

From the point of view of when errors are detected, we distinguish:

1.1. Compile Time Error-Syntax errors and static seman-tic errors indicated by the usage of the compiler. Compile time errors are resolved manually.

1.2. Runtime Error-Dynamic Semantic mistakes and log-ical error that can't be detected by way of compiler.

Programming Errors are mainly classified into three cate-gories on the basis of formulation of errors-

1) Syntax Error-The syntax of the language is not respected.
2) Semantic Error-Semantic errors are stated via the compiler while the statements written within the c software are not significant to the compiler.
3) Logical Error-The specification is not respected.

2. Debugging

Runtime mistakes are resolved by using debugging.Debugging is the procedure of detecting and removing of existing and capability errors in a software program code that could motive it to act suddenly or crash. To save you incorrect operation of a software or machine, debugging is used to find and remedy bugs or defects. When diverse subsystems or modules are

ISBN : 978-93-87793-78-1

tightly coupled, debugging will become more difficult as any alternate in one module may additionally reason greater bugs to seem in some other. Sometimes it takes extra time to debug a application than to code it.

3. Code Refactoring

Code refactoring:smooth your code is the process of restructuring current pc code—changing the factor-ing—without converting its outside conduct.

Refactoring is a controllable process of improving code without creating new functionality. It transforms a

The process of changing a software program machine in this type of manner that it does no longer regulate the external conduct of the code but improves its inner structure mess into easy code and easy layout.

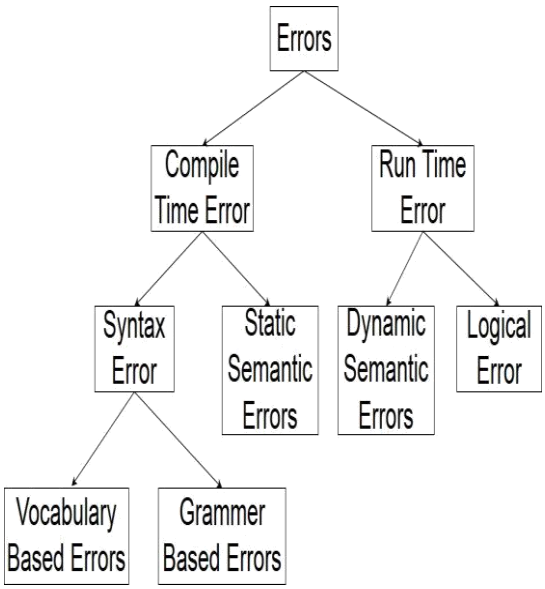

Fig 1: Types of Errors

4. Code Refactoring

Code refactoring:smooth your code is the process of restructuring current pc code—changing the factor-ing—without converting its outside conduct.

Refactoring is a controllable process of improving code without creating new functionality. It transforms a

The process of changing a software program machine in this type of manner that it does no longer regulate the external conduct of the code but improves its inner structure mess into easy code and easy layout.

5. Code reformatting

Based on user definable code styles.As a user I want to be able to easily format code based on my personal preferences so my code is easier to read and navigate.

6. Autocompletion

Basic code of completion allows you whole names of training, strategies, fields, and keywords inside the visibility scope.

Autocomplete or phrase completion works so that after the writer writes the first letter or letters of a word, the program predicts one or extra possible phrases as choices. If the word he intends to write is included in the list he can select it.

III. PROPOSED METHOD

Students who are new to programming are not aware of programming language so they used to write incorrect code. So the proposed approach consist of data preprocessing module which will assist the code by correcting the errors made by students.

Proposed appoach consist of following modules-

1. Data Preprocessing

Method will take incorrect code which is written by student as input and preocess it line by line if it has incorrect data or not.

Input file will be taken with following syntax:

ifstream file(”test.cpp”);

Output file will be created as follows:

outputfile.open(”clonetype.cpp”,ios::out);

1.1 Lexical Phase Errors: Lexical error is a sequence of characters that does not match the pattern of any token. Lexical phase error is found during the execution of the program.

- Spelling error.

ISBN : 978-93-87793-78-1

- Exceeding length of identifier or numeric constants.
- Appearance of illegal characters.
- To remove the character that should be present.
- To replace a character with an incorrect character.

1.2 Syntactic Phase Errors: During the syntax analysis phase, this type of error appears. Syntax error is found during the execution of the program.

1.3 Semantic Errors: Semantic errors arise for the duration of the execution of the code, after it has been parsed as grammatically correct. These ought to do no longer with how statements are built, but with what they imply. Such things as incorrect variable kinds or sizes, nonexistent variables, subscripts out of variety, and the like, are semantic mistakes.

Algorithm 1: Programming Assistant

1: Check for header file
2: if not correct then
3: Replace the header file with the maximum matching in the header database
4: Check for main function
5: if not written correctly then
6: Correct the name of main function
7: Check for open curly braces of main function
8: if not present then
9: put one after the main function
10: if datatype not written correctly then
11: replace them with the maximum matching datatype from the database
12: Store all the declared variable in an list.
13: if keyword not written correctly then
14: replace that keyword with the maximum matching keyword from the keywords database
15: if undeclared variable is present then
16: declare that variable if not present in list of declared items
17: Keep the track of all the opening and closing curly braces separately
18: Apply semicolon at the end of the line if not present other than conditions and loops
19: Balance all the double quotes
20: Balance all the parenthesis
21: if for() then Statecorrect the syntax of for loop if not written correctly.
22: if open curly braces >closed curly braces then
23: balance the curly braces by applying the remaining closed braces.
24: Close the curly braces of if condition before else.

2. Compile Time Errors

2.1 Header file-

Header files play a very important role in running code.A header record is a document with extension H which incorporates C function declarations and macro definitions to be shared among numerous supply documents. There are two sorts of header documents: the documents that the programmer writes and the documents that comes together with your compiler.

2.2 Keyword Correction-

Keywords are predefined, reserved words utilized in programming which have unique meanings to the compiler. Keywords are part of the syntax and they can't be used as an identifier.

For example: int temp;

2.3 Double quotes of i/o-

Double quotes of printf anf scanf are very important for taking input and displaying output in in scrren.

2.4 Semicolon Placement- Semicolon at the end of sentence(not after condition and loops) is very important. Without semicolon compiler will generate a error of missing semicolon.

2.5 Bracket balance-

Brackets on every line should be balanced to produce a correct output without any error.

ISBN : 978-93-87793-78-1

2.6 Datatype Name Correction-

Name of datatype should be written correctly becouse each datatype have different ranges. So to produce correct output name of the datatyupe should be written correctly.

2.7 Syntax within for loop-

syntax within for loop means proper placing of semi-colon in for for loop syntax.

The initialization assertion is executed simplest as soon as.

Then, the test expression is evaluated. If the take a look at expression is fake (zero), for loop is terminated. But if the test expression is proper (nonzero), codes inside the body of for loop is carried out and the replace expression is up to date.

This process repeats till the take a look at expression is fake.

2.8 Curly braces of if condition- curly braces of if condition should be closed before else for proper execution of the program.

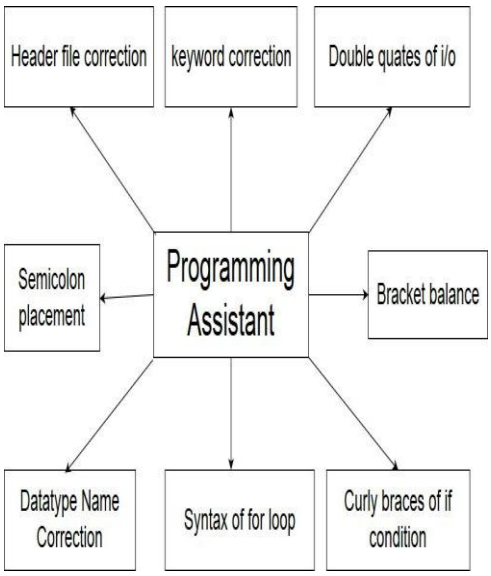

Fig. 2: Extension of Compile Time Errors

3. Correcting Errors

3.1 Header Files: In our work,we do start with correcting header files. Header files are the basis of any c program. To use any function in c,one should use a correct header file. So here we correct all the header files.

We will store all the header files in a file

After storing all the header files in an array(header), we will complare the writtten string with each of the header file and with which ever header file, the string matched the most, the string will get replaced.

we will replace the incorrected header file with the max-imum matching correct header file, like #include<stio>will be converted to #include<stdio.h>.

3.2 Main function: First we check that the line contain main function related string or the header file.if the line contains header file related string then correction will be done according to header file correction and if line contains main function related string then that string will be replaced with main().

we will replace every string written in the place of main function with main(). Simply main() will be placed in the place of wrong syntax.

3.3Grammer Correction:

- Main function braces

Most of the time students forget to close the braces of main function.

Curly braces of main function will automatically get balanced if not present at the end of the program. Braces will be closed at the end of the program.

- Double quates of i/o function

If double quates are not applied in the right places then Double quates of printf and scanf will get balanced automatically. Sometimes when we need to print the entire predefined synax then Double quates for printf will be closed at the end of the sentence or we can say before the brackets.

ISBN : 978-93-87793-78-1

- Curly braces of if condition

Curly braces of if condition will be closed before else.If else isn't present then it will simply balance the braces as other braces ar getting balanced.

- Balance of all the curly braces

Many a times Students forget to close the curly braces. If any curly braces are missing then compiler will show an error of missing curly braces. So if any curly braces is missing then in our work all the curly braces, which are missing, will get balanced.

- Semicolon at the end of sentence

In the starting phase,most people make an error with putting semicolon at the right places, they usually put comma, fullstop and colon in the place of semicolon. So semicolon will get apllied at the end of the sentence(if needed).

- Brackets("(",")")

If some where in the program, brackets are missing then at compile time it will generate an error of brackets missing. For that error, one has to the location of the error and make the brackets balanced. But in our work, All the brackets will get balanced automatically.

4) **Names of the keywords:** keywords which are already predefined in c library, if someone write those keyword incorrectly then those incorrect keywords will be corrected with the nearly matched keyword.

5) **Name of the Datatypes**: Many a times name of the datatype are written incorrectly by mistake. So in our work, Name of all the datatypes will get corrected automatically. Names will replaced with the nearly matched datatype.

6) **Syntax of for loop:** Following are the steps of for loop-

6.1 The init step is finished first, and only as soon as. This step permits you to claim and initialize any loop manipulate variables. You aren't required to position a declaration right here, so long as a semicolon appears.

6.2 Next, the situation is evaluated. If it's far genuine, the frame of the loop is achieved. If it's miles fake, the body of the loop does not execute and the float of manipulate jumps to the following statement just after the 'for' loop.

6.3 After the frame of the 'for' loop executes, the waft of manage jumps back up to the increment announcement. This assertion permits you to update any loop control variables. This statement can be left blank, as long as a semicolon appears after the condition.

All the steps will be seperated with the semicolon but most of the time students makes an error with putting semicolon at the right places. Putting semicolon in between the above steps is a very essensial thing otherwise compiler will continue to show an error with the proper placing of the semicolon.

Most of the time students make mistake with the syntax of for loop means they put comma and colon in the place of semicolon. Syntax with in the for loop will get corrected means semicolon will be applied on the right places.

7) **Undeclared variables:** Sometimes student use vari-ables which aren't declared, so at that time compiler show an error like undeclared variable.

So in this study undeclared variable will automatically detected and get declared.

IV. TESTING AND RESULTS

Proposed approach is implemented using C++ language.

It takes incorrect code written by students as input.

Errors which have been corrected are I/O operaton, Header file correction, Main function correction, Braces of main() function, Braces of if condition, Syntax of if

else, Syntax within for loop, All the curly braces, Undeclared Variable.

A module for every error is implemented and all the module are merged into a single method which will correct all the described errors.

Following is the study which has been done to find the error which mostly made by the students-

In this research work a study has been done on 1st year student(350 students approximately) in their programming labs. In that study, it has been seen that most of the time students are just stuck on the compile time errors and are not able to develop their logical ability towards programming. Errors which are made by students are as following-

1) Incorrect header file
2) Incorrect main function
3) Incorrect name of the datatypes.
4) incorrect keywords.
5) Use of variables which are not declared.
6) Grammar errors in Programming-which are as follows-
   - Curly braces of main function.
   - Missing semicolon at the end of the line other then conditions and loops.
   - Imbalanced curly braces.
   - Curly braces of ”if” condition are not close before else.
   - Double quotes of ”printf” and ”scanf” are not balanced.
   - Syntax of for loop like semicolon at the right places is not present.
7) Use of variables which are not declared.

The percentage of students who make a perticular type of errors which is as follows-

1) Grammatical Errors are made by 60-70% students.So Correction of these errors is important because a major part of the class is making mistake with this syntax.
2) 30-40% students made error with writting key-word,datatype and header files.
3) 20-25% student write incorrect main function.

1. **Analysis of the Result**

Errors which have been described will automatically get corrected. Every errors will get corrected by it’s own module.

After developing the programming assistant tool, It has been tested on 2nd semester student(Approximately 350 students).

Errors which are solved are as follows-

1) Approximately 90% of grammatical error means spe-cial symbols will get corrected.
2) All the header files will get corrected means 100% errors generated with writting of header files.
3) 100% of datatype and keywords errors will get cor-rected.
4) 100% errors with name of main file will get corrected.

V. CONCLUSION

By developing programming assistant tool in C-programming, Students will be able to write program more effectively and easily. It also increases interaction of students to the programs. They can better understand the program and read ,write it. Removing syntax errors will increase the readability of the code by which student will be able to read the code snippets as well. Students will better understand the program logic rather then being stuck on the syntax error or we can say compile time errors.

ISBN : 978-93-87793-78-1

REFERENCES

[1] J. C. MATHES, MEMBER, IEEE, AND KAN CHEN,Educational gg Objectives for Science, Technology, Society, and Values Programs,IEEE TRANSACTIONS ON EDUCATION, VOL. E-21, NO. 1, FEBRUARY 1978.

[2] Cliff Lampe,””EDUCATIONAL PRIORITIES FOR TECHNOLOGY-MEDIATED SOCIAL PARTICIPATION””, 0018-9162/10 2010 IEEE .

[3] Susan Lysecky and Jerzy Rozenblit,””Educational Technologies for Precollege Engineering Education”,”, IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, VOL. 5, NO. 1, JANUARY-MARCH 2012.

[4] Sonia Pamplona, Isaac Seoane, Javier Bravo-Agapito, and Nelson Medinilla ,””Insights into Students’ Conceptual Understanding of Op-erating Systems: A Four-Year Case Study in Online Education””, IEEE Communications Magazine November 2017.

[5] ASHISH DUTT1, MAIZATUL AKMAR ISMAIL1, AND TUTUT HERAWAN,””systematic review on educational data mining””, ADig-ital Object Identifier 10.1109/ACCESS.2017.2654247.

[6] M. Borrego, A.Chan Hilton, L.J. Everett, S. Finger, D. Millard,””New National Science Foundation opportunities for improving undergradu-ate engineering education””, IEEE conference on education,19 Decem-ber 2013.