

Common Mistakes in Writing Basic Elements of C++ Programming for Dummies

Syarifah Adilah Mohamed Yusoff, Rozita Kadar and Saiful Nizam Warris
syarifah.adilah@uitm.edu.my, rozita231@uitm.edu.my, saifulwar@uitm.edu.my

Jabatan Sains Komputer & Matematik (JSKM), Universiti Teknologi MARA Cawangan Pulau
Pinang, Malaysia

Introduction to writing a program

“Understanding that in the future there is no profession is untouched by machines means admitting that coding is part of the liberal arts” according to Wall Street Journal (2015). Therefore, giving basic education on programming language to primary and secondary school is one of strategic planning in Education 5.0. Not every child learn writing will become a novelist, nor everyone who learn algebra a mathematician, yet we treat both as foundation skills for all children. Coding is the same since the future consist of emerging industries of cryptocurrency and Artificial Intelligence space. Coding is uniquely suited to training children not just how to solve problems, but also how to express themselves (Resnick, 2019).

Introduction to programming language has been taught to several programmes in university level. In Universiti Teknologi MARA Cawangan Pulau Pinang, Malaysia, engineering students for both diploma and degree levels are compulsory to enrol the course. The necessity of taking this course lies on practising problem solving skills, expose students to writing programming language and simulate creative thinking in solving problems. All of the students are categories as novice programmers due to has no official learning both in programming and computer essential. Hence, learning computer language is not only new but difficult to understand the rules and structures. The next sections will discuss about common mistakes in basic elements of programming and input and output statements respectively.

Common mistakes on basic elements of C++

Mistakes or errors in writing C++ program could happen in three ways, the most common one is syntax errors, followed by run-time errors and lastly, logical errors. This section will discuss thoroughly some common errors happens when the basic concept of programming is concerned.

Declaration is most basic elements, yet crucial statement in writing a program. In general, students easily understand the concept of declaring variables and the four rules of naming variables. Understand the concept of data in a program, memory allocations and how CPU and computer memory run the source code are some of important piece of information that have to be clarified to the novice programmers. The explanation would help them understand the foundation of why declarations is important in a program and visualise how the processor and memory works in order to execute the declaration instructions. In the next subsections, several common errors of the basic concept in C++ will be discussed.

Confusion of data representation of variables

In C++, students are well known about variations of data types such as `int`, `float`, `double`, `long`, `char`, `string` and `bool`. Apparently, when a case study is given most of novice programmers will become confused with the data store in the variables and the variable itself.

Example 1.0

*Write a program to mimic a simple calculator. Prompt user to enter two number and an operation of +, -, /, *. The program will display the result of the selected operation.*

Given the Example 1.0, this type of question required programmer to do selection in the program based on operation selected by user. Three data are required in order to calculate and get the output. Students have to analyse these three data in order to declare variables with appropriate data types. Two of the data are numeric data, meanwhile the confusing part is to represent the third data which is operator. The most occurred error is declaring all the operators

as variables such as `char +, -, /, *`; without considering the valid variable's name guided by the naming rules of identifiers. The right way is students have to understand that `+, -, /` and `*` are data that user has to key in and select only one of them. Therefore, a valid variable with character data types is required to store the selected operator such as, `char option;`.

Example 2.0

The Entrepreneur Club would like to offer canopies to be rented on Entrepreneur Week celebration. As programmer you have been appointed to write a program to calculate the total charge of rental for the whole week. The charges are based on category of traders entered by user whereas; student (T) is charge RM 95.99; Staff(S) is charge RM150.99; and Outsider(O) is charge RM200.99 all in a day.

Another Example 2.0 is considering to select code of T, S or O in order to calculate the rental charge of the canopy. The most common error done by students is declaring variable as `char T, S, O;` which shows the inability to differentiate between data and appropriate variable. Students suppose understand that user will select the data of T or S or O when they key in and store it into the allocation of memory based on the proper variable declaration. Example of valid declaration is `char tradecode;` this indicate a variable name `tradecode` is declared as character data type. The `tradecode` is supposed to store any valid data of T or S or O entered by user using appropriate input statement.

Choosing suitable data types

In a program, apparently data is the most important element, where data will be manipulated and transformed into information for output display. Students has to write instructions to allocate memory location for the data when the program is running. The instruction is called declaration statements which consist of variables and appropriate data types. The data types are reserved word in C++ or any programming languages. Students has to know the suitable data types for that particular data used in the program.

When the data is numeric that come with unit of measurements such as kg, cm, litre, or currency symbol such as RM, students intend to initialised the data together with the units. Here are example of invalid declaration and initialisations, `int price_sugar= RM10;` `float volume= 25kg;` `double length=34.7cm;` all the unit mistakenly considered as part of the data. The data should only consider the raw values that required for further

manipulation. The units of measurement or currency are treated as label and not the raw data. Program can output or display the labels anytime using output statements. Thus, the declaration and initialisation only consist of raw data with appropriate data types such as, `int price_sugar= 10; float volume= 25;double length=34.7;`

Mismatch declarations with input statement

In general, data entered into a program either with initialisation or prompt input from user. The concept of request data from user is the most flexible way to get fresh data. In solving real life problems, most of the data is collected directly from users. The second method is retrieved data from database, anyhow this method is not covered for fundamental study in C++ course syllabus.

The instruction for getting the data from user into a program is constructed as input statements. The input statement has to work closely with the declaration statements since the process of entering data require memory allocation in order to store the data for further manipulation. C++ input statement usually begin with `cin` and followed by stream extraction operator `>>` and next followed by dedicated variable of the data. The syntax is `cin>>variable1>>variable2; .` This input statement only valid for numeric and single character data. When user has to enter alphabet data, there are two input statements has to be considered and depends to the declaration statements of the dedicated variables. There are two data types involves which are `char` and `string`, examples of declaration are `char patients_n[30];` and `string address1; .` The first declaration, `char patients_n[30]` means a variable name of `patients_n` is declared, whereas memory location is set and labelled as the variable's name with the maximum size of characters are 30. The second declaration, `string address1;` means a variable `address1` is declared to store alphabet data types. Appropriate input statements for both data are `cin.getline(patients_no, 30);` and `getline(cin, address1);` subsequently.

Students tends to forget the input statement for `string` and `char` data type. They usually continue repeat the errors until the end of semester. Table 1.0 is summarized the basic concept of implementing declarations, input statements and closely related with difference categories of data types adapted from Mohamad & Mydin (2019).

Table 1. Association of basic elements of C++.

Categories of data	Data Types	Description of data	Size Byte	Declaration statement	Initialisation	Input Statement
Numeric	float	Decimal numbers	4	float no1, no2;	no1=2.8; no2=10.7;	cin>>no1; cin>>no2;
	double		8	double _a;	_a=2.8973;	cin>>_a;
	int	Whole numbers	4	int a,b,c;	a= 100, b= 5, c=	cin>>a>>b >>c;
	long	Both decimal and whole numbers	8	long int x;	x=452122;	cin>>x;
Character	char	One character	1	char code;	code='T';	cin>>code ;
		Limited characters with max size	>1	char name[30];	name= 'a','b', 'c'; name="abc" ;	cin.getli ne(name,3 0);
Alphabet	string	Small, capital letters, symbols, numerics	>8	string status;	status="" available" ;	getline(c in, status);
Boolean	bool	True/1 (as long as the value is NOT 0), False/0	2	bool delicious;	delicious= true; or delicious= 1;	cin>>deli cious;

Common mistakes on input/output statements

Syntax for input statement is very straight forward, students have to understand the solely purpose of input statement is to prompt data from user. Therefore, the syntax of input statement is based on the `cin>>` and followed by variables such as explained in previous section. Meanwhile output statements are used to display various variables' values, arithmetic expression and any "message in between double quote". The examples of various purposes output statements are shown in Table 2.0.

Table 2. Variation of output statements

Purpose	Example of output statements
Display variable's value	<pre>int num=2;//appropriate variable declaration cout<<num;</pre>
Display arithmetic expression	<pre>float a=3.2; //appropriate variable declaration cout<<a*5;</pre>
Display messages on output screen	<pre>cout<< "\n welcome to C++"; cout<< "3 +9 = ";</pre>

When writing a program, input and output statements are usually written together in order to create user friendly instruction. As an example, to prompt user to enter weight and height, the basic instruction is using `cin>>` statement as Example 3.0 below:

Example 3.0:

```
float weight, height;  
cin>>weight>>height;
```

Anyhow, as for user friendly instructions we may add messages before user entered data. This will help user to give accurate data for the program purpose. The appropriate instruction is in Example 4.0 as follows:

Example 4.0:

```
float weight, height;  
cout<< "Enter weight in KG:";  
cin>>weight;  
cout<< "Enter height in feet:";  
cin>>height;
```

Here, when the commands `cin>>` and `cout<<` are used together, most of students tends to get confuse with the usage and the purpose of both commands. The wrong instructions are shown in Example 5.0 and Example 6.0 as follows:

Example 5.0:

```
float weight, height;  
cout<< "Enter weight in KG:"<<weight;  
cout<< "Enter height in feet:"<<height;
```

or

Example 6.0:

```
float weight, height;  
cin>>"Enter weight in KG:">>weight;  
cin>>"Enter height in feet:">>height;
```

Example 5.0, shows no syntax errors in both `cout<<` statements. Misunderstood of message "Enter weight in KG:" and variable `weight` displayed after the message as input statement is almost happen for those ignorance students to the concept of input and output. It is already clear the roles of `cin>>` statement to enter data and `cout<<` statement to display data. Therefore, Example 5.0 will display the messages and the value of variables `weight` and `height` if any.

Example 6.0 will cause syntax errors when compiling the statements. Students have to remember the purpose of each statements. The purpose to write messages "Enter weight in KG:" and "Enter height in feet" are to display the messages on output screen, therefore the right command in C++ is `cout<<` statement. Meanwhile variable `weight` and `height` are used to retrieve data right after user key in into computer. Undoubtedly, the `cin>>` statement is required here.

Conclusion

Basic elements of programming cover the essential of commands for declaring, initialisation, input and output the data. The information is fundamental every time the new program is developed. If students unable to master the concept, they won't able to write a free error program and facing more difficulties in writing complicated program in future chapters.

References:

Mims, C.(2015, April 26).Why Coding Is Your Child’s Key to Unlocking the Future. *Wall Street Journal*. Retrieved from <https://www.wsj.com/articles/why-coding-is-your-childs-key-to-unlocking-the-future-1430080118>.

Resnick, M.(2019, Jan 3). The Next Generation of Scratch Teaches More Than Coding. *EdSurge*. Retrieved from <https://www.edsurge.com/news/2019-01-03-mitch-resnick-the-next-generation-of-scratch-teaches-more-than-coding>.

Mohamad,W, A & Mydin, A. (2019). *Introduction to C++ Programming(2nd).* Selangor: Oxford Fajar Uni. Press.