UTM UNIVERSITI TEKNOLOGI MALAYSIA OPENCOURSEWARE

SSCM 1313 C++ COMPUTER PROGRAMMING

Chapter 1: Introduction to Programming Language

Authors:
Farhana Johar
Professor Dr. Shaharuddin Salleh

BY NC SA
innovative • entrepreneurial • global
ocw.utm.my

Chapter 1

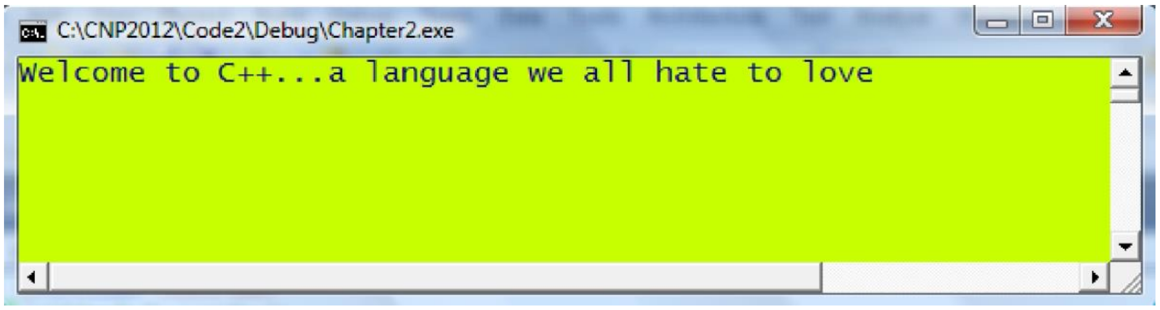
C++ is a general-purpose programming language that evolves in the early 1980's. This language extends from the popular C language whose procedural approach became a model to other languages in its class. C does not support the object-oriented style of programming, but this deficiency is overcome through its extension into C++. With this extension, C++ now supports both the object-oriented as well as the procedural type of programming.

Definition 1.1: *Procedural programming* involves breaking down the problem into several procedures or modules or functions, where a solution to the problem is obtained by solving these modules.

Definition 1.2: *Object-oriented programming* involves breaking down the problem into some entities called *objects*, where a solution to the problem is obtained through a series of control and manipulation over these objects.

Code1A.cpp: The very first C++ program.

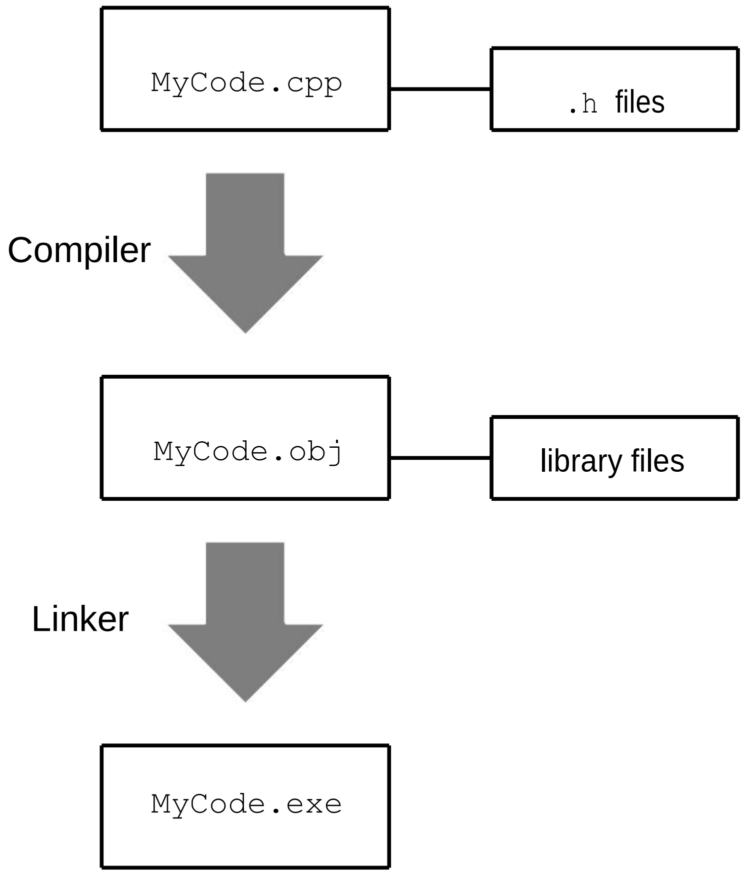
```
#include<iostream>

using namespace std;

void main()
{
   cout << "Welcome to C++ ...a language we all hate to love" << endl;
   cin.get();
}
```


C:\CNP2012\Code2\Debug\Chapter2.exe

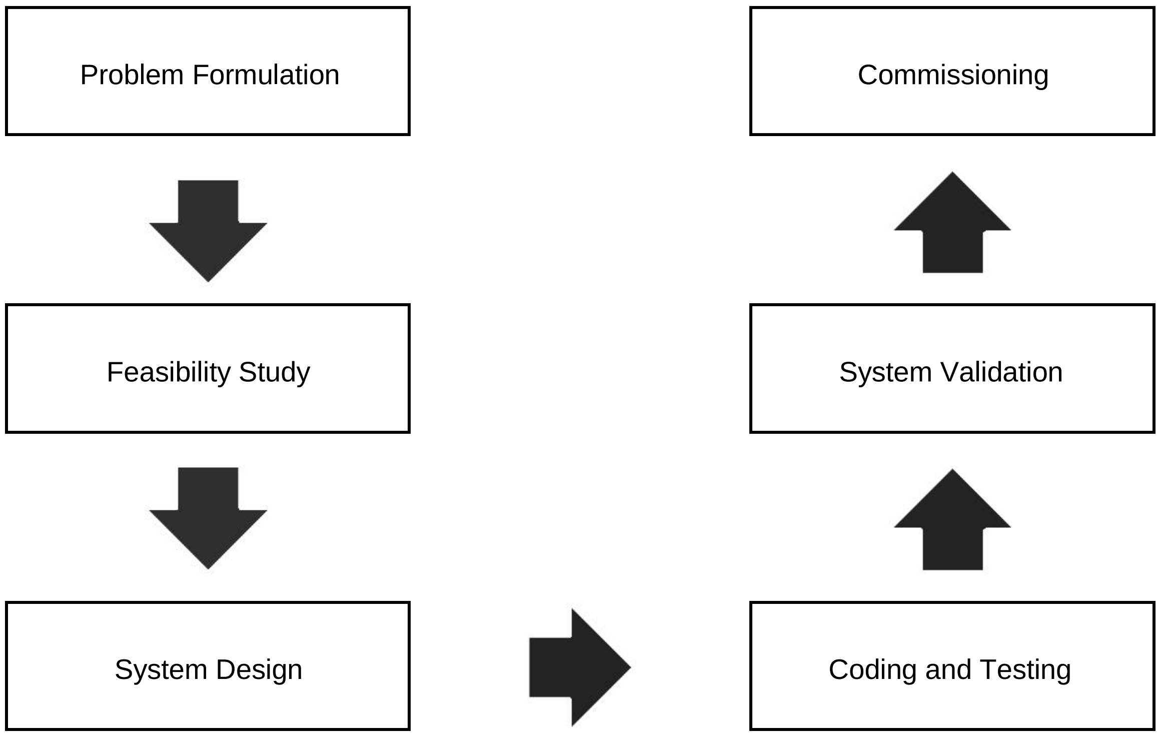
```
Welcome to C++...a language we all hate to love
```

statement	*A single and complete instruction in the program which ends with the symbol ;*
directive	*An instruction to the compiler for controlling the program, for example, to include a header file.*

Compile, Link and Go



Program Design

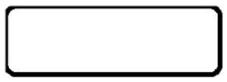


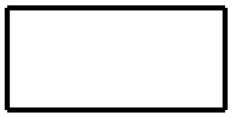
Algorithm

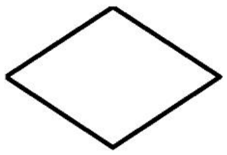
$$z = \frac{1 - 3\sum_{i=1}^{10} x_i^{\;2} y_i}{4\sum_{i=1}^{5} x_i^{\;2} + 7\sum_{i=1}^{7} y_i^{\;2}}$$

```
Read the equation;
Read the input data;
Find the numerator value;
Find the denominator value;
if the denominator is 0
     No solution, so abandon the operation;
else
     Divide the numerator by the denominator;
     Display the result;
End if
```

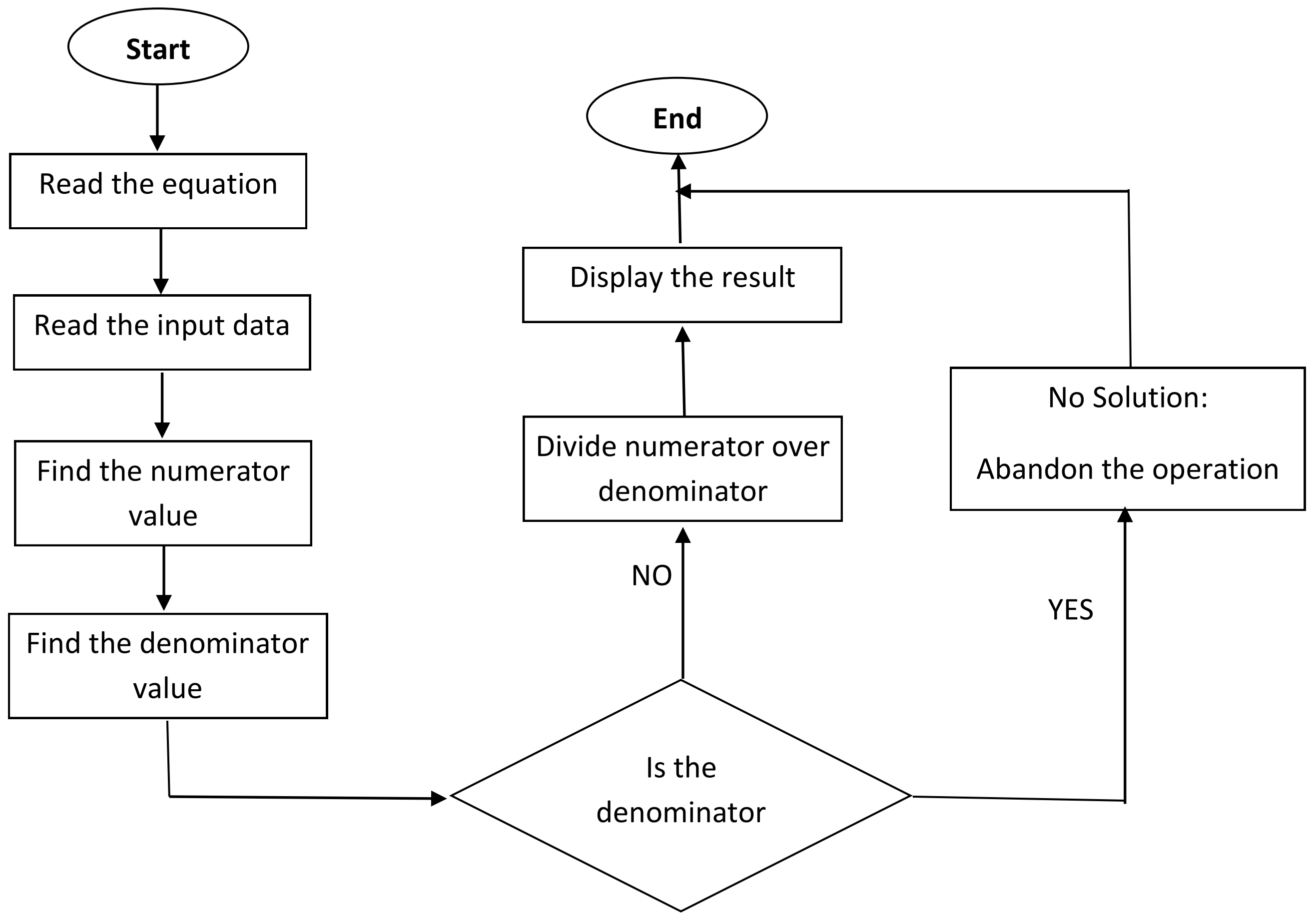
Flowchart


Indicates the start or end of the program.


Indicates an assignment.


Decision making which indicates a test for conditional branching of TRUE or FALSE.

UTM UNIVERSITI TEKNOLOGI MALAYSIA OPENCOURSEWARE



Pseudocode

```
Read the values of $x_i$ and $y_i$
Find $u = \sum_{i=1}^{10} x_i^2 y_i$ , $v = \sum_{i=1}^{5} x_i^2$ and $w = \sum_{i=1}^{7} y_i^2$
Find $p = 1 - 3u$ , $q = 4v + 7w$
if $q = 0$
        No solution, so abandon the operation.
else
        Find $z = \frac{p}{q}$
        Display the result
End if
```

innovative • entrepreneurial • global ocw.utm.my

Best Practices in C++

- Make a complete feasibility study to the problem before starting its design. Decide whether C++ programming is necessary in providing the solution to the problem in terms of costs, time, human factor and viability.
- In general, any application that has numerical solutions can be solved using C++. Make sure the numerical solution to a problem exists by solving a simple version of the problem using a calculator.
- The designer should have the thorough knowledge in the concepts and solution to the problem. Plan ahead. Develop algorithms, flowcharts and pseudocodes before writing the codes.
- Start writing the program in a small scale by working on a simple version of the problem. Then slowly add one component to the problem at a time, until it reaches the desired size. In that way, the program becomes manageable and easy to modify.
- Be object-oriented by organizing the variables, structures and functions into classes.
- Be structured and modular in the program design.
- Declare the correct data types for variables and functions. Don't use too many variables unnecessarily. Also, maximize the use of local variables in order to make the functions autonomous.
- Use a lot of structures for relating the variables.
- Avoid redundant statements. Use a lot of loops instead of repeating statements.
- Limit the size of every function by dividing the task independently into several functions.
- Make the program scalable for supporting any size of data without the need to revamp the whole program codes massively.
- Provide a complete documentation in the program design so that it can be referred later easily.

UTM UNIVERSITI TEKNOLOGI MALAYSIA OPENCOURSEWARE

Modeling and Simulation

Simulation can be implemented using three approaches: microscopic, macroscopic and mesoscopic [1]. In the *microscopic* simulation, the detail physical and performance characteristics, such as the properties of the elements that make up the problem are considered. The simulation involves some tiny properties of the individuals or elements that make up the pieces. The results from a microscopic simulation are always reliable and accurate. However, this approach could be very costly and time consuming as data from the individuals or elements are not easy to obtain.

An easier approach is the *macroscopic simulation* which considers the deterministic factors of the whole population, rather than all the individuals or elements. In this approach, factors such as the governing mathematical equations and their macro data are considered. The steps in this approach may skip the detail components, and, therefore, the results may not be as accurate as the one produced in the microscopic approach. However, this approach saves time and is not as costly as the microscopic approach.

A more realistic and practical approach is the *mesoscopic simulation* which combines the good parts of the microscopic and macroscopic approaches to produce a more versatile model. In this approach, some deterministic properties of the elements in the system are integrated with the detail information to produce a workable model such as in Bacteria Population Growth, Computational Fluid Dynamics, Finite Element Modeling, Printed-Circuit Board Design, Wireless Sensor Networks or GPS Navigation System.

MAIN REFERENCE:

Shaharuddin Salleh (2012), C++ Numerical Programming.

BY NC SA