Introduction to Programming
Lecture No. 1 & 2

Muhammad Salman

ABASYN UNIVERSITY

What is programming

As this course is titled "Introduction to programming," it is essential and appropriate to understand what programming really means.

"A program is a precise sequence of steps to solve a particular problem."

Practically, in our day to day lives we are constantly planning, organizing and paying attention to fine details (if we want our plans to succeed). And it is also fun to do these activities. For example, for a picnic trip we plan where to go, what to wear, what to take for lunch, organize travel details and have a good time while doing so.

Computer Programming

When we talk about computer programming then to put it simple:

“At its most basic level, programming a computer simply means telling it what to do”

The first hard thing about programming is to learn, become comfortable with, and accept these artificial mechanisms, whether they make ``sense'' to you or not. “

Why Programming is important ?

The question most of the people ask is why should we learn to program when there are so many application software and code generators available to do the task for us.

“The answer consists of two parts. First, it is indeed true that traditional forms of programming are useful for just a few people. But, programming as we the researcher understand it is useful for everyone:

Hence learning to program is important because it develops analytical and problem solving abilities. It is a creative activity and provides us a mean to express abstract ideas.

Thus programming is fun and is much more than a vocational skill. By designing programs, we learn many skills that are important for all professions. These skills can be summarized as:

Tools of the trade?

As programmers, we need different tools to develop a program. These tools are needed for the life cycle of programs.

Editors

First of all, we need a tool to write the code of a program. For this purpose we used Editors in which we write our code Text editors are editors that save only the text which we type. So for programming, we will be using a text editor.

Compiler and Interpreter

As we write the code in English and, we know that computers can understand only 0s and 1s. So we need a translator which translates the code of our program into machine-readable language.

There are two kinds of translators which are known as Interpreter and Compilers. These translators translate our program which is written in C++ into Machine language.

Interpreters

Interpreters translate the program line by line meaning it reads one line of the program and translates it, then it reads the second line, translates it, and so on. The good part is that we get the errors as we go along and it is very easy to correct the errors. The drawback of the interpreter is that the program executes slowly as the interpreter translates the program line by line.

Compiler and Interpreter

Compilers also translate the English-like language (Code written in C++) into a language (Machine-readable language) that computers can understand. The Compiler reads the whole program and translates it into machine language completely.

The difference

The difference between the interpreter and compiler is that the compiler will stop translating if it finds an error and there will be no executable code generated where as the Interpreter will execute all the lines before the error and will stop at the line which contains the error. So, the Compiler needs a syntactically correct program to produce an executable code.

First C++ program

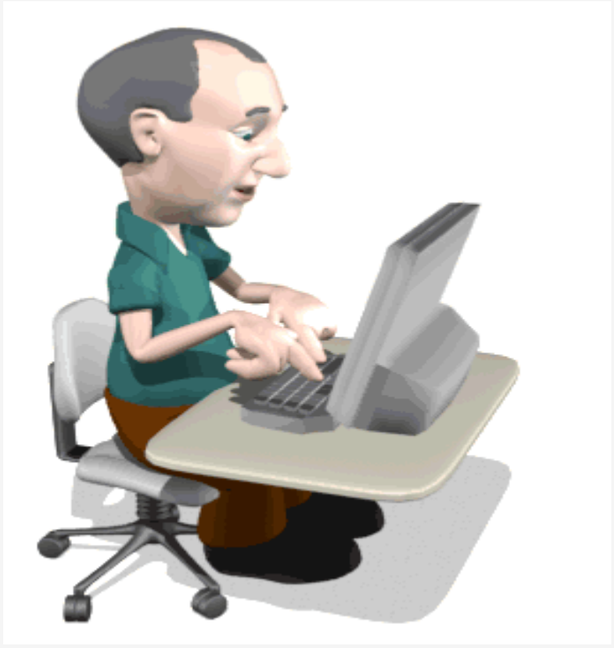
main.cpp

```cpp
// Our First C++ Program

#include <iostream>
using namespace std;

int main() {
    cout << "Welcome to Abaysn Universiy";
    return 0;
}
```

Output

```
Welcome to Abaysn Universiy

=== Code Execution Successful ===
```

The same Program but different syntax



```cpp
// Our First C++ Program

#include <iostream>
//using namespace std; commented/Disable

int main() {
  std:: cout << "Welcome to Abaysn Universiy";
    return 0;
}
```

Output

```
Welcome to Abaysn Universiy

=== Code Execution Successful ===
```

The order of the C++ Statements matters !!!

```cpp
#include <iostream>
using namespace std;


int main() {
    // Write C++ code here
    cout << "   / |"<<endl;
    cout << "  /  |"<<endl;
    cout << " /   |"<<endl;
    cout << "/____|"<<endl;
    cout<<"it prints Top to Buttom\n";
    cout<<"last line";

    return 0;
}
```

Output

```
   / |
  /  |
 /   |
/____|
it prints Top to Buttom
last line
```

Underspending the Program (Syntax)

Let's look at this code line by line and try to understand it.

 # include <iostream.h>

- #include: This is a pre-processor directive.
- It is not part of our program but an instruction to the compiler.
- It tells the C++ compiler to include the contents of a file, in this case, the system file iostream.h
- These are also known as Compiler Directives.

<iostream.h>

- This is the name of the library definition file for all Input Output Streams.
- Our program will almost certainly want to send stuff to the console(<<cout)and read things from the keyboard (>>cin).
- iostream.h is the name of the file which has code to do that work for

Underspending the Program (Syntax)

main()

- The main() function indicates the beginning of every C++ program.
- The main() must be included in every C++ program.
- When a C++ program is executed, the control goes directly to the main() function.
- If the main() function is not included in the C++ program it is not compiled and an error message will be generated.

C++ Statement { }

- The program's statement is written under the main() function between the curly braces { }.
- These statements are the body of the program.
- Each statement in the C++ ends with semicolon ;

Underspending the Variable

Variable

- During programming, we need to store data. This data is stored in variables.
- Variables represent a storage or memory location in the computer memory.
- Data is stored in the memory location.
- The name of the memory location i.e. the variable name remains fixed during the execution of the program.
- Think of it as PO Boxes. In post offices, there are different boxes and each has an address. Similarly in memory, there is a numerical address for each location of memory (block).
- We call them variables because they can contain different values at different times.

Underspending the Variable

- A variable name must start with a letter or the underscore character.
- A variable name cannot start with a number.
- Variable names are case-sensitive (name, Name, and NAME are three different variables).
- The reserved words(keywords) cannot be used to name the variable in C++.

Some Basic Datatypes

The data type specifies the size and type of information the variable will store:

```
int myNum = 5;                  // Integer (whole number without decimals)
float myFloatNum = 5.99;     // Floating point number (with decimals)
char myLetter = 'D';           // Character
string myText = "Hello";      // String (text)
bool myBoolean = true/false;         // Boolean (true or false)
```

Data Type	Size	Description
`boolean`	1 byte	Stores true or false values
`char`	1 byte	Stores a single character/letter/number, or ASCII values
`int`	2 or 4 bytes	Stores whole numbers, without decimals
`float`	4 bytes	Stores fractional numbers, containing one or more decimals. Sufficient for storing 6-7 decimal digits
`double`	8 bytes	Stores fractional numbers, containing one or more decimals. Sufficient for storing 15 decimal digits

Some Basic Datatypes

int Data Type (integer) int x=4;

- The data type int is used to store whole numbers (integers).
- The integer type space of 4 bytes (32 bits for Windows operating system) in memory.
- it is worth mentioning as 'int' which is a reserved word of C++, so we can not use it as a variable name.

float Data Type float y = 4.1

- This value is represented by the float class. It is a real number with a floating-point representation. It is specified by a decimal point.
- The float data type uses four bytes to Store a real number. Here is a program that uses float data types.

double Data Type double y = 4.1

- If we need to store a large real number that cannot be stored in four bytes, then we use a double data type.
- Normally the size of a double is twice the size of a float. In the program, we use it as: double x = 345624.769123;

Some Basic Datatypes

char Data Type char z='A'

- In programming, we do need to store characters like a,b,c, etc.
- For storing the character data C++ language provides char data type.
- Using char data type we can store characters in variables.
- While assigning a character value to a char type variable single quotes are used around the character as 'a'.

Strings string a = "Welcome to the Abasyn University"

- A string is a collection of one or more characters Strings are used for storing text/characters.
- A string variable contains a collection of characters surrounded by double quotes
- To use strings, you must include an additional header file in the source code, the <string> #include <string>

Operators

- Operators are symbols with defined functions.
- Programmers use these symbols to tell the interpreter or compiler in high-level computer languages, such as C++, Java, and Python, to perform a particular action.
- These symbols form the program's foundation, allowing you to perform various actions ranging from simple math to complex encryption.

- Operators are essential for performing calculations, assigning specific values to variables, and making condition-based decisions.
- We will learn more about operator types and how they work in C++.

Types of operators

- Different types perform different tasks within the program.
- There are three main types of operators, each with unique functions and capabilities.

Arithmetic operator

Arithmetic operators allow computers to perform mathematical calculations on specific values. For example, we might write 'A+B = 40' or 'A*B = 100'. Common arithmetic operators include:

- Addition: +
- Subtraction: -
- Multiplication: *
- Division: /
- Integer division: DIV
- Remainder: MOD

C++ Program using arithmetic operators

Write a C++ program to perform the arithmetic operation using all arithmetic operators and print the result on the screen.

Method #1

```cpp
//write a C++ Program to perform the arithmetic Operation by using all,
//arithmetic operator and print the result on the screen.
#include <iostream>
using namespace std;

int main() {
  int a,s,m,d,r;

  //float d1= 5.0/c;
  a= 5+2;
  s=5-2;
  m=5*2;
  d=5/2; ///float
  r=5%2;
  cout <<"Addition of 5 & 2 is  =    "<<a<<endl;
  cout <<"Subtraction of 5 & 2 is  =    "<<s<<endl;
  cout <<"Multiplication of 5 & 2 is  =    "<<m<<endl;
  cout <<"Division of 5 & 2 is  =    "<<d<<endl;
  cout <<"Modules/Reminder of 5 & 2 is  =    "<<r<<endl;
  //cout <<"The float division is     = "<<d1<<endl;
  return 0;
}
```

Output

```
Addition of 5 & 2 is  =    7
Subtraction of 5 & 2 is  =    3
Multiplication of 5 & 2 is  =    10
Division of 5 & 2 is  =    2
Modules/Reminder of 5 & 2 is  =    1
```

Problem statement:

But the problem with this program is that we have to enter the number “2” manually but we it to assign it to a variable and perform all the arithmetic operations.

C++ Program using arithmetic operators

Write a C++ program to perform the arithmetic operation using all arithmetic operators and print the result on the screen.

Method #2

```cpp
#include <iostream>
using namespace std;
int main()
{
int a,s,m,d,r;
   a=s=m=d=r=5;
  int z=2;
  a=a+z;
  s = s-z;
  m=m*z;
  r=r%z;
  cout <<"The addition of a and z = " <<a<<endl;
  cout << "The subtraction a and z =  " <<s<<endl;
  cout << "The multiplication a and z =  " <<m<<endl;
  cout << "The modulas a and z =  " <<r<<endl;

return 0;

}
```

Output

```
The addition of a and z = 7
The subtraction a and z =  3
The multiplication a and z =  10
The modulas a and z =  1
```

Method #2 Explanation:

In this program, we have initialized/declared variables a,s,m,d, and r and assigned 5 as a value using the assignment operator =. We did the same for variable z, which has a value of 2, and then performed operations such as a=a+z = a= 5+2.

Types of operators

Relational operator

Relational operators facilitate condition testing, allowing us to create and assign variable values. For example, if A equals 45 and B equals 50, you might write A < B or A is less than B. That < symbol is a relational operator that produces true or false results. Familiar relational operators include:

- Assignment = int A = 45, B = 50
- Equivalence ==
- Less than <
- Greater than >
- Less than or equal to <=
- Greater than or equal to >=
- Does not equal !=

Types of operators

Logical operator

Logical operators allow programmers to gain increasingly complex computer-based decisions by combining relational operators. For example, with the logical AND operator, if A equals 10 and B equals zero, and you write 'A AND B is false,' the condition is true if both variables are the same. If you use the logical OR operator and just one of the variables is something other than zero, 'A OR B is true' would be true.

You can reverse the variable's logical state with the local NOT operator. So, what would otherwise be true becomes false, etc.

Common operators include:

- AND (&&)
- OR (| |)
- NOT (!)

C++ Program

```cpp
#include <iostream>
#include <string>
using namespace std;

int main() {

    string a ;
    string b;
    cout<<"Enter your first Name here : "<<endl;
    cin>>a;
    cout<<"You have entered: " <<   a <<endl;
    cout<<"Enter last Name: "<<endl;
    cin>>b;
    cout<<"You have entered "<< b<<endl;
    cout<<"Your full name is : " <<  a.append (b);


}
```

Output

```
Enter your first Name here :
salman
You have entered: salman
Enter last Name:
muhammad
You have entered muhammad
Your full name is : salmanmuhammad
```

C++ Program

```cpp
#include <iostream>
#include <string>
using namespace std;

int main() {

    string a ;
    string b;
    cout<<"Enter your first Name here : "<<endl;
    cin>>a;
    cout<<"You have entered: " <<   a <<endl;
    cout<<"Enter last Name: "<<endl;
    cin>>b;
    cout<<"You have entered "<< b<<endl;
    cout<<"Your full name is : " <<  a.append (" ").append(b);


}
```

Output

```
Enter your first Name here :
Muhammad
You have entered: Muhammad
Enter last Name:
Salman
You have entered Salman
Your full name is : Muhammad Salman
```

C++ Program

```cpp
#include <iostream>
#include <string>
using namespace std;

int main() {

    string f_name ;
    string l_name;
    cout<<"Enter your first Name here : "<<endl;
    cin>>f_name;
    cout<<"You have entered: " <<   f_name <<endl;
    cout<<"Enter last Name: "<<endl;
    cin>>l_name;
    cout<<"You have entered "<< l_name<<endl;
    cout<<"Your full name is : " <<  f_name.append (" ").append(l_name);


}
```

Output

```
Enter your first Name here :
Muhammad
You have entered: Muhammad
Enter last Name:
Salman
You have entered Salman
Your full name is : Muhammad Salman
```

C++ Program

Write a C++ Program to read the temperature in Fahrenheit. Convert the temperature to Celsius degrees by using the formula c=(f-32)5/9 or c=(f-32)1.8.

C++ Program

Write a C++ Program to read the student's name and marks obtained in the three subjects C++, Operating Systems, and Machine Learning. Calculate the total and average marks. Using "cin" object.

```cpp
using namespace std;

int main() {
    string os ;
    string  ml;
    int osmarks;
    int mlmarks;
    int result;
    float avg ;

    cout << " Enter subject name: "<< endl;
    cin >> os ;
    cout <<"Enter marks : "<<os<<endl;
    cin >> osmarks;
    cout <<"Enter Subject name : "<<endl;
    cin >> ml;
    cout<<"Enter marks " <<ml<<endl;
    cin>>mlmarks;
    result = osmarks+mlmarks;
    avg = result/2.0;
    cout <<avg;

    return 0;
}
```

C++ Program

Write a C++ Program to compute and print the volume of a cylinder where its radius "r" an Height "h" are given using cin object **(Hint : vol= πhr^2 and π = 3.14)**

Increment & Decrement Operator

Increment:

The Increment operator used to add 1 to the value of a variable is called the increment operator.

The increment operator is represented by a double plus (++) sign. It is used to add 1 to the value of an integer variable. This operator can be used before the variable name.

xy = xy+1

```cpp
// Online C++ compiler to run C++ program online
#include <iostream>
using namespace std;

int main() {
    int a = 5;
    int b=5;
    int x=5;
    int sum = a+b + (x++);

    cout <<"The sum of a+b+x = ;" <<sum<<endl;
    cout <<"The valus of x is now = " <<x;
     return 0;
}
```

Output

```
The sum of a+b+x = ;15
The valus of x is now = 6
```

Increment & Decrement Operator

Decrement (--):

The Decrement operator is represented by a double minus (--) sign. It is used to subtract 1 from the values of an integer variable.

For example, to subtract 1 from the values of a variable xy the decrement statement is written as xy– or x--

xy = xy-1

```
#include <iostream>
using namespace std;

int main() {
   int a = 5;
   int b=5;
   int x=5;
   int sum = a+b + (--x);

   cout <<"The sum of a+b+x = " <<sum<<endl;
   cout <<"The valus of x is now = " <<x;
    return 0;
}
```

Output

```
The sum of a+b+x = 14
The valus of x is now = 4
```

Thanks