PROGRAMMING FUNDAMENTALS LAB

Course code: CS118

Credit hours: 1

Date: 04-Sep-2019

RECAP

- Questions from previous lecture?

https://en.wikipedia.org/wiki/Camel_case

OBJECTIVES

- Learn basic components of a C++ program.
- Learn the compilation process.
- Learn how to use *Visual Studio*
- Learn how to write, compile, and run simple C++ programs.
- Learn basic data types in C++.
- Learn how to use data type in problem solving.
- Learn how to declare a variable and Constants

STRUCTURE OF A C PROGRAM

- Preprocessor Directives that always begin with #
  - Example: **#include <iostream>**

  The two most frequently used directives are **#include** and **#define**
- **#include** directive will include header files that have the definitions of functions used in the program.
  Example: **cout** function is defined in the header file **iostream**
- **#include <*file*>** tells the compiler to look for *file* where system include files are held
- **#include "*file*"** tells the compiler to look for *file* in the current directory where the program was run from
- **#define** is used to replace a text with a value, e.g. **#define PI 3.141593**

CONT'D…

- **Functions**
- A function is a block of statements that perform a specific kind of processing. Every C program has a main() function
- The program execution starts from main()

- **Variables / identifiers**
- These are used to store data in a program. You must declare a variable before you use it in a program

GENERAL STRUCTURE OF A C++ PROGRAM

peprocessor directives

main function heading

{

declarations

executable statements

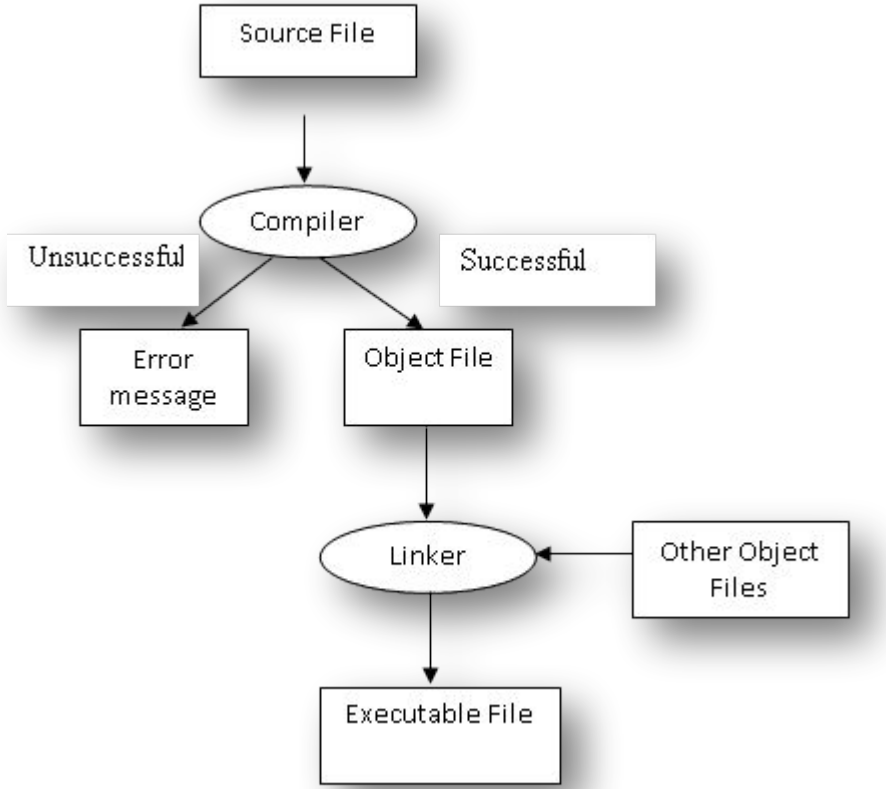
}

```
#include <iostream>
using namespace std;
int main(void)
{

cout<<”Asalam o Alaykum”;
return 0;
}
```

COMPILATION PROCESS

- The process of converting a source file (Example: A C++ program) into a machine language program



VARIABLES (INTRODUCTION)

- By its definition, ***a variable is a place in memory that holds some information***.
  - Some information may be numbers, characters, strings or Booleans

- **Variables declaration:**
- *Every variable must be declared before use*
  - Before using a variable, you must first let the compiler know
- The compiler will need two pieces of information concerning each variable: the **amount of space** the variable will need, and **a name to recognize** that variable.
- **type variableName;**

VARIABLE NAMES

The name of a variable:

- Starts with an underscore "_" or a letter, lowercase or uppercase, such as a letter from a to z or from A to Z. Examples are Name, gender, _Students, pRice.
- Cannot include special characters such as !, %,], or $
- Cannot include an empty space
- Cannot be any of the reserved words
- Should not be longer than 32 characters (although allowed)

Note: C++ is **case-sensitive**; this means that CASE, Case, case, and CaSe are four completely different words.

BASIC DATA TYPES

Data Type	Keyword	Bytes	Range	Example Data
Character	char	1	-128 to 127	A
Integer	Int	4	-2,147,483,648 to 2,147,483,647	1
Floating Point	float	4	-3.4E38 to 3.4 E38 (appx. 7)	1.1
Double precession floating point	double	8	-1.7e308 to 1.7e+308 (appx. 15)	1.0000000008

MULTIPLICATION OF 2 NUMBERS

```
#include <iostream>
using namespace std;
int main() /* begins program execution */
{
    int num1,num2;
      num1 = 15;
    num2 = 20;
    int result = num1 * num2;
    cout<<num1<<"*"<<num2<<"="<<result;
    return 0;
}
```

SQUARE OF A NUMBER

```
#include <iostream>
using namespace std;
int main()
{
int num = 50;
cout<<“The number”<<num;
cout<<”Square of number”<< num*num;
return 0;
}
```

AREA AND CIRCUMFERENCE OF A CIRCLE

```
#include <iostream>
using namespace std;
# define PI  3.14159
int main()
{
    double radius, area, circumference;
    radius = 5.0;
    area = PI * radius * radius;
    circumference = 2 * PI * radius;
    cout<<"Radius = “<<radius;
    cout<<"Area = “<<area;
    cout<<”Circumference = “<<circumference;
    return 0;
}
```

DISPLAY A VALUE OF A VARIABLE

```
#include <iostream>
using namespace std;
int main ()
{
    int age;     //declaration
    cout<<"This value of age is “<<age;
    return 0;
}
```


Why
What
Where
Who
When