Name of Teacher: Miss Radhika M. Patil

Class: B.Sc. Computer Science (Entire)- II **Semester : 3**

Course Title: Object Oriented Programming using C++

Execution of C++ program:

For creating and executing a C++ program, one must follow four important steps.

(1) Creating Source code

Creating includes typing and editing the valid C++ code as per the rules followed by the C++ Compiler.

(2) Saving source code with extension .cpp

After typing, the source code should be saved with the extension **.cpp**

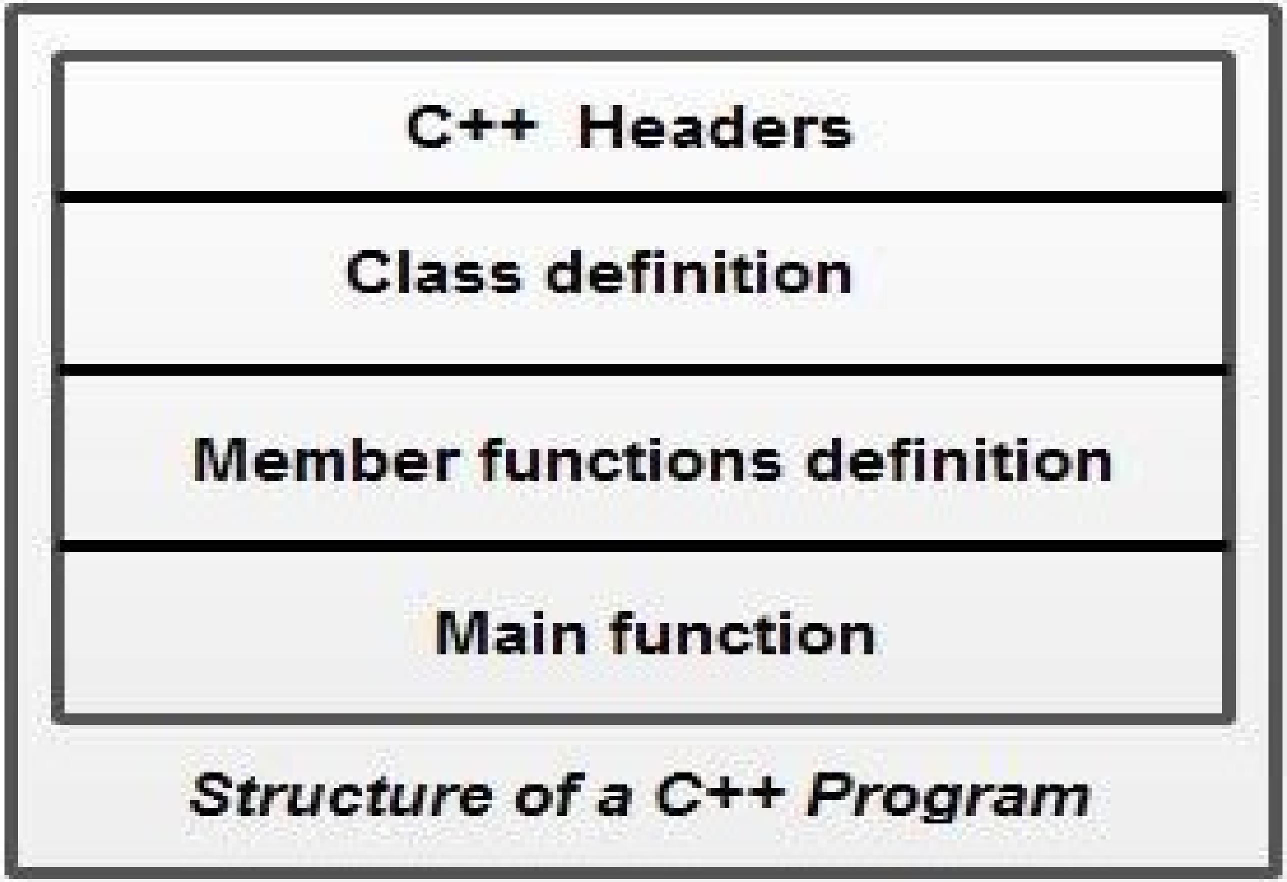
(3) Compilation

This is an important step in constructing a program. In compilation, compiler links the library files with the source code and verifies each and every line of code. If any mistake or error is found, it will inform you to make corrections. If there are no errors, it translates the source code into machine readable object file with an extension **.obj**

(4) execution

This is the final step of construction of a C++ Program. In this stage, the object file becomes an executable file with extension **.exe**. Once the program becomes an executable file, the program has an independent existence. This means, you can run your application without the help of any compiler .

Structure of C++ Program



Structure of a C++ Program

Documentation

/*...*/ OR // comments are used for the documentation to understand the code to others. These comments are ignored by the compiler.

1. **C++ Headers:**

   It contains pre-processor statements to include various header (library) files.

2. **Class Declaration:**

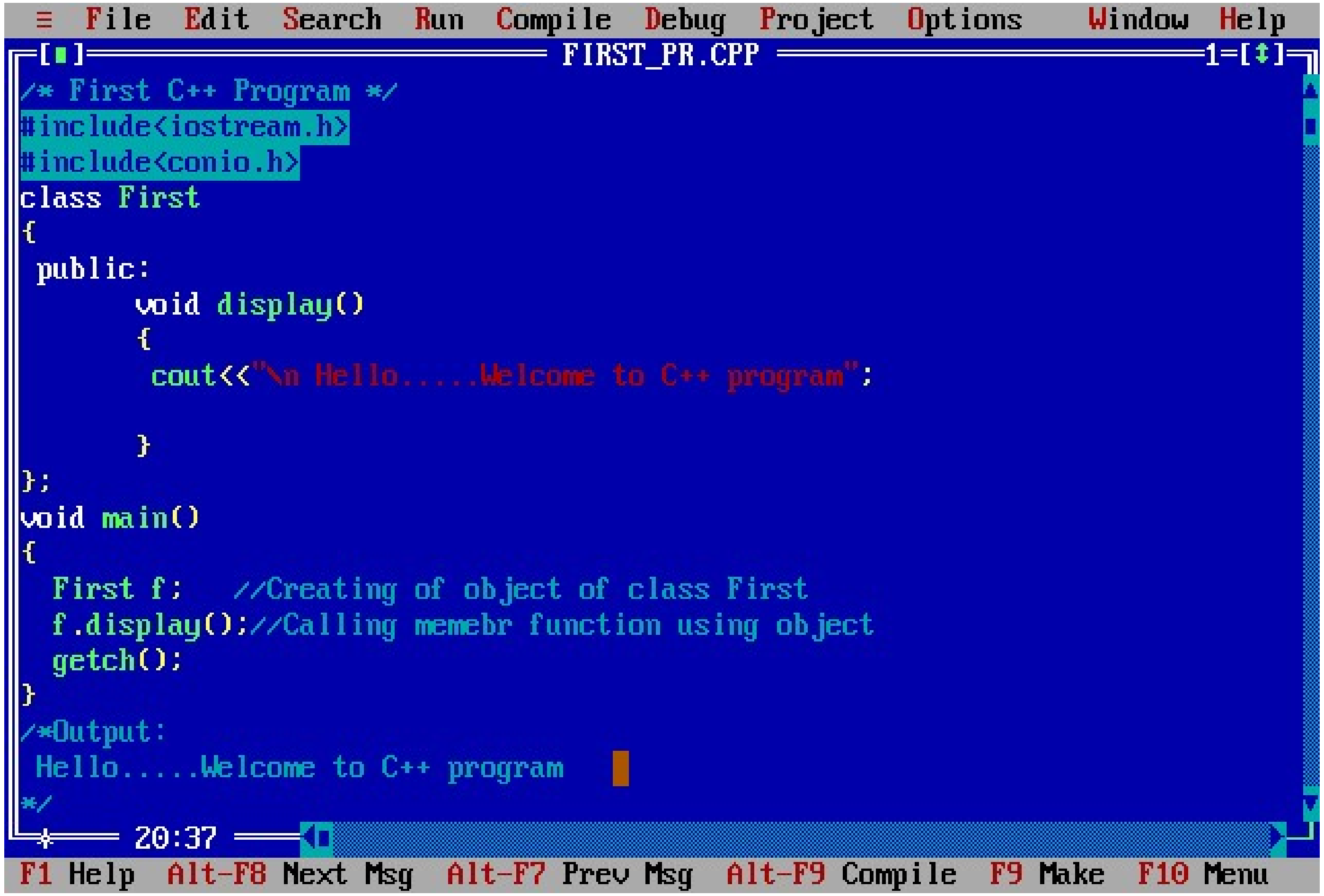
   After the include section there is class declaration section. In class declaration we are defining class, its member variable and declaration for the member function.

   Class member function definition section contains the definition of various class member functions.

3. **Main program:**

   At last there is a main function, which start the execution of program and create the object of class.

   Main program can use more than one class similarly one class can be used for more than one program.


≡ File Edit Search Run Compile Debug Project Options Window Help
FIRST_PR.CPP
1
20:37
F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu

```
/* First C++ Program */
#include<iostream.h>
#include<conio.h>
class First
{
 public:
       void display()
       {
        cout<<"\n Hello.....Welcome to C++ program";

       }
};
void main()
{
  First f;     //Creating of object of class First
  f.display();//Calling memebr function using object
  getch();
}
/*Output:
 Hello.....Welcome to C++ program
*/
```

I/O stream link
https://www.slideshare.net/nileshdalvi01/input-and-output-in-c-39132148

Memory management operators

- Arrays can be used to store multiple homogenous data but there are serious drawbacks of using arrays.
- Programmer should allocate the memory of an array when they declare it but most of time, the exact memory needed cannot be determined until runtime.
- The best thing to do in this situation is to declare the array with maximum possible memory required (declare array with maximum possible size expected) but this wastes memory.
- So, To **avoid wastage of memory**, you can dynamically allocate the memory required during runtime using new and delete operator.

What are memory management operators?

There are two types of memory management operators in C++:

- new
- delete

These two memory management operators are used for allocating and freeing memory blocks in efficient and convenient ways.

New operator:
The new operator in C++ is used for dynamic storage allocation. This operator can be used to create object of any type.
General syntax of new operator in C++:
pointer variable = new datatype;

In the above statement, new is a keyword and the pointer variable is a variable of type datatype.
For example:
1. int *a = new int;
2. *a = 20;
or
3. int *a = new int(20);
In the above example, the **new** operator allocates sufficient memory to hold the object of datatype int and returns a pointer to its starting point.
the pointer variable a holds the address of memory space allocated.

delete operator:

The **delete** operator in C++ is used for **releasing memory** space when the object is no longer needed.
Once a new operator is used, it is efficient to use the corresponding delete operator for release of memory.

General syntax of delete operator in C++:

delete pointer variable;

THANK YOU…