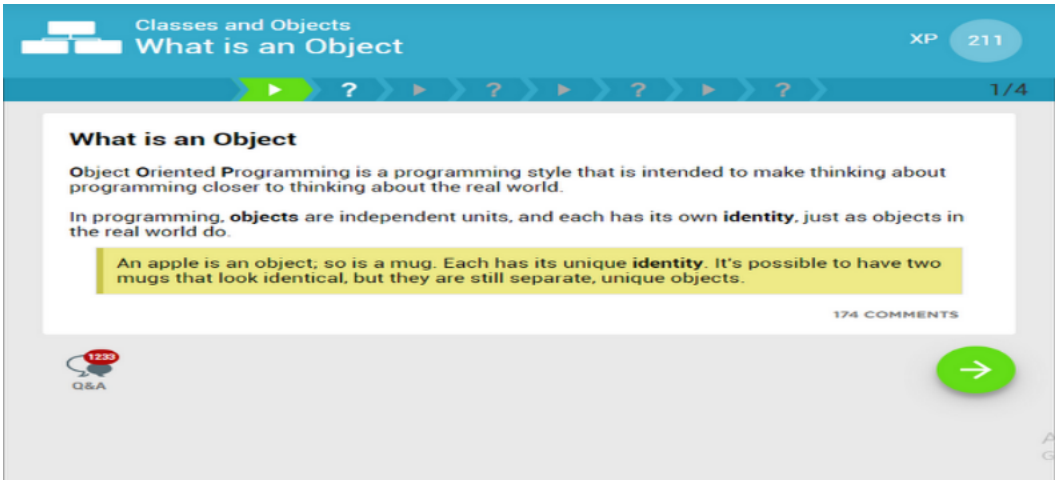
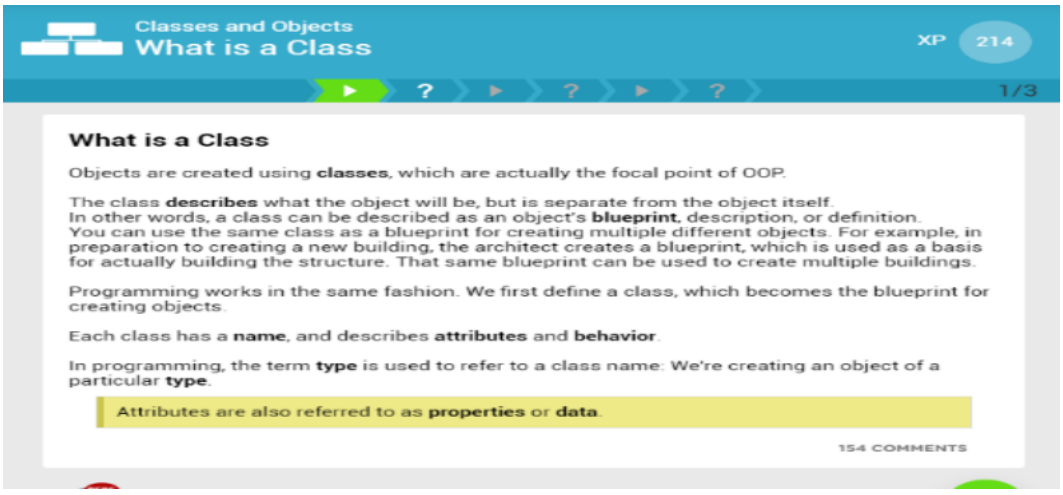
DAILY ASSESSMENT FORMAT

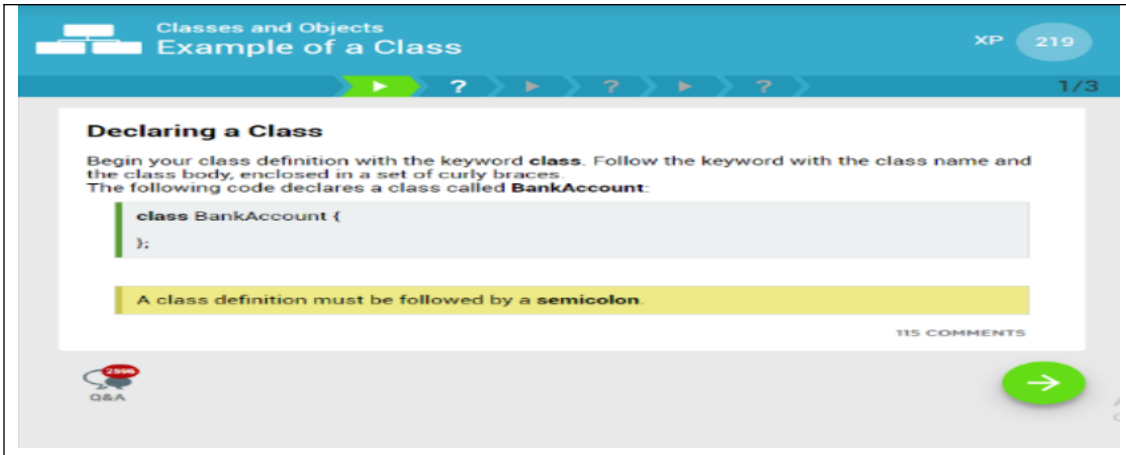
Date:	24-06-2020	Name:	Rajeshwari Gadagi
Course:	C++ Programming	USN:	4AL17EC076
Topic:	Module 5	Semester & Section:	6th SEM & 'B' SEC
Github Repository:	Rajeshwari-gadagi		

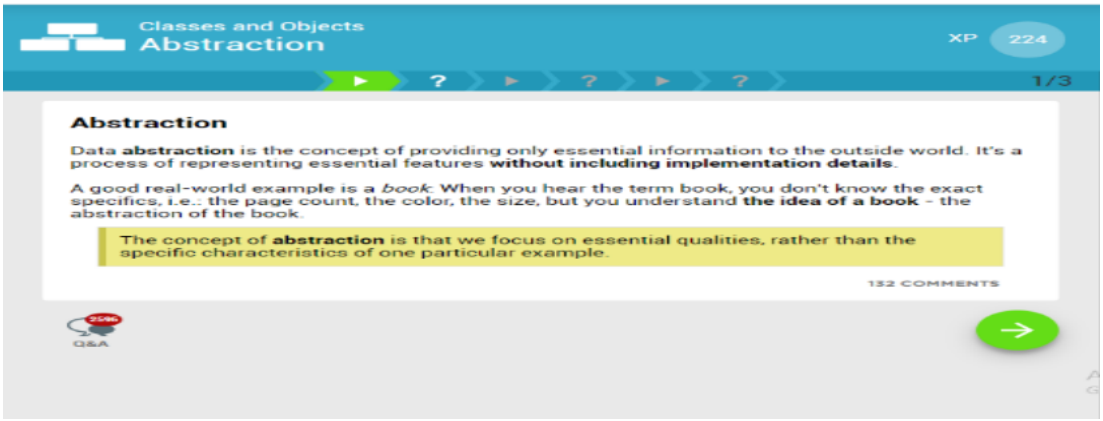
FORENOON SESSION DETAILS

Image of session









Report – Report can be typed or hand written for up to two pages.

C++ Programming

24/04/2020 - Wednesday.

Modules:

-> Classes & Objects

> What is an Object?
Object Oriented Programming is a programming style that is intended to make thinking about programming closer to thinking about the real world.

* An Object might contain other objects but they are still different objects

* In programming, an Object is self-contained, with its own identity. It is separate from other Objects.

> What is class
* Object are created using classes, which are actually the focal point of OOP.

* Each class has a name, & describes attributes & behaviour.

* Methods : Method is another term for a class behaviour. A method is basically a function that belongs to a class

> Declaring a Class

* Begin your class definition with the keyword class.

* Declaring a class:
Define all attributes & behaviour in the body of the class, within curly braces.
You can also define an access specifier for members of the class

* Creating a class

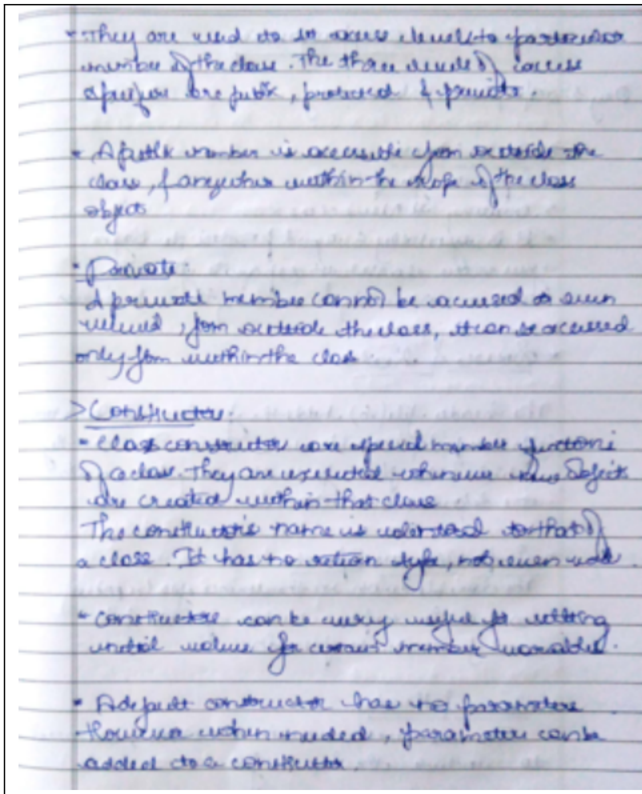
> Abstraction :
* Data abstraction is the concept of providing only essential information to the outside world. It's a process of representing essential features without including implementation details.

* Abstraction means, that we can have an idea or a concept that is completely separate from any specific instance

> Encapsulation :
Part of the meaning of the word encapsulation is the idea of "surrounding" an entity, not just to keep what's inside together, but also to protect it.

> Example of Encapsulation

- Access Specifiers



• They are used to set access levels to particular member of the class. The three levels of access specifier are public, protected & private.

• A public member is accessible from outside the class, for anything within the scope of the class object.

• Private:
A private member cannot be accessed or even viewed from outside the class, it can be accessed only from within the class.

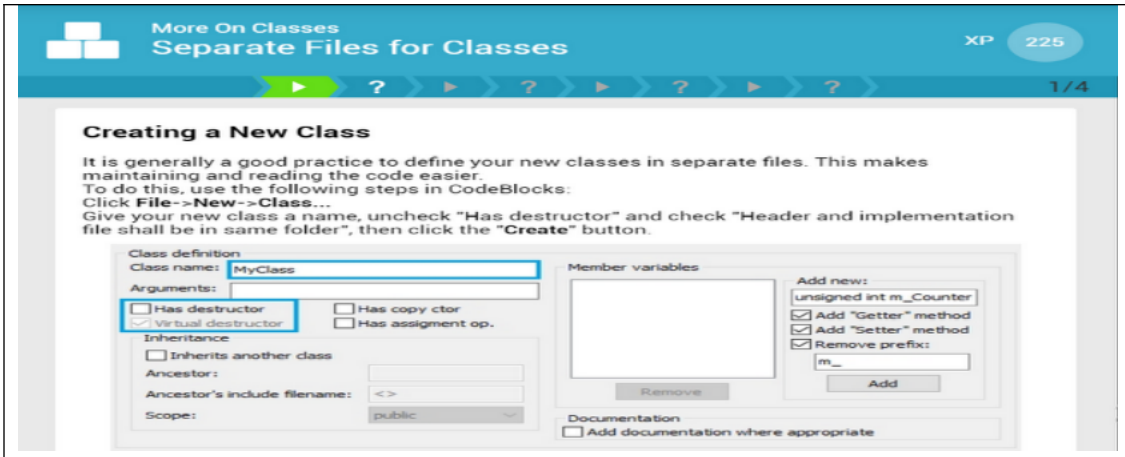
> Constructor:
• Class constructors are special member functions of a class. They are executed whenever new objects are created within that class.
The constructor's name is identical to that of a class. It has no return type, not even void.

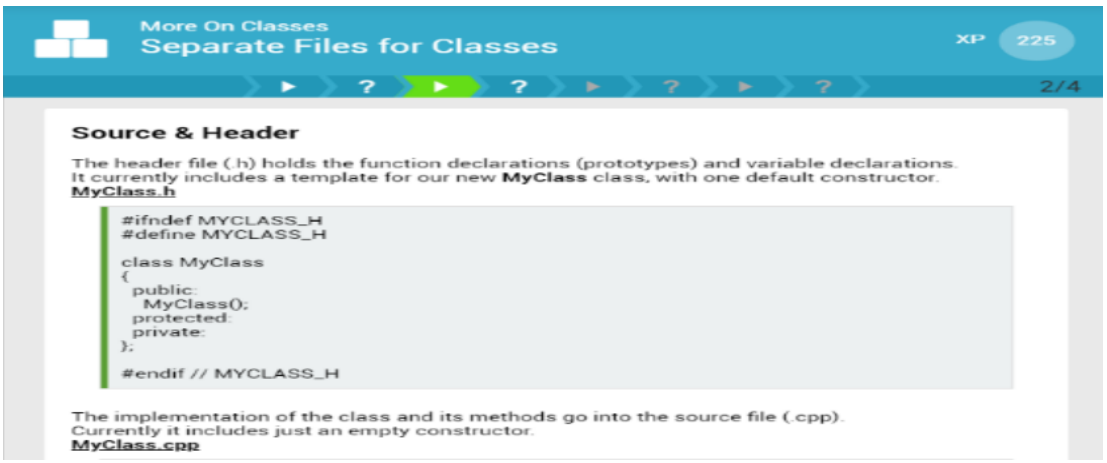
• Constructors can be very useful for setting initial values for certain member variables.

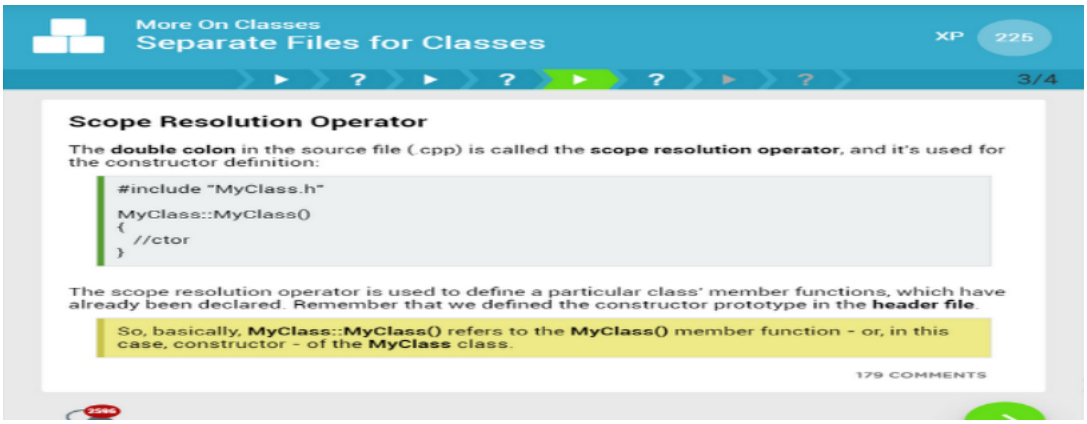
• A default constructor has no parameters. However when needed, parameters can be added to a constructor.

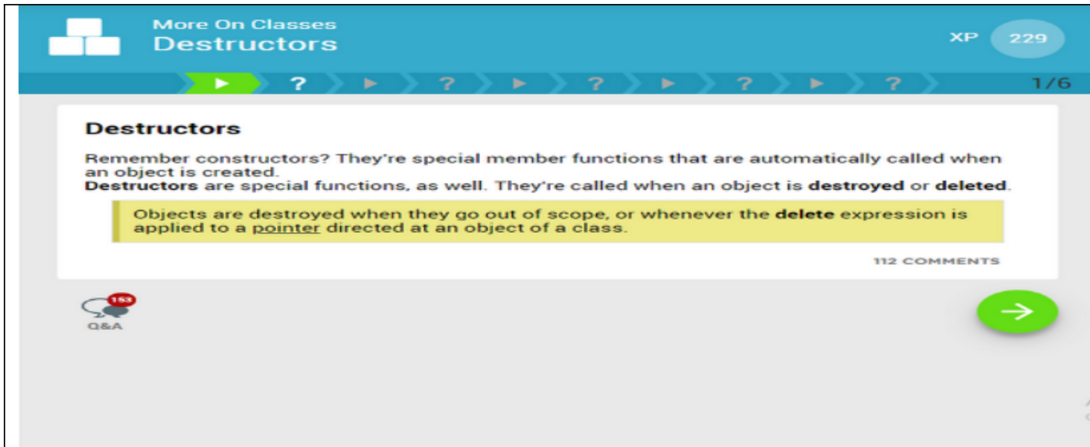
Date:	24-06-2020	Name:	Rajeshwari Gadagi
Course:	C++ Programming	USN:	4AL17EC076
Topic:	Module 6	Semester & Section:	6 SEM & 'B' SEC
Github Repository:	Rajeshwari-gadagi		

AFTERNOON SESSION DETAILS
Image of session


More On Classes
Separate Files for Classes
XP 225
1/4
Creating a New Class
It is generally a good practice to define your new classes in separate files. This makes maintaining and reading the code easier.
To do this, use the following steps in CodeBlocks:
Click File->New->Class...
Give your new class a name, uncheck "Has destructor" and check "Header and implementation file shall be in same folder", then click the "Create" button.
Class definition
Class name: MyClass
Arguments:
Has destructor
Virtual destructor
Has copy ctor
Has assigment op.
Inheritance
Inherits another class
Ancestor:
Ancestor's include filename: <>
Scope: public
Member variables
Add new:
unsigned int m_Counter
Add "Getter" method
Add "Setter" method
Remove prefix:
m_
Add
Remove
Documentation
Add documentation where appropriate


More On Classes
Separate Files for Classes
XP 225
2/4
Source & Header
The header file (.h) holds the function declarations (prototypes) and variable declarations.
It currently includes a template for our new MyClass class, with one default constructor.
MyClass.h
#ifndef MYCLASS_H
#define MYCLASS_H
class MyClass
{
public:
MyClass();
protected:
private:
};
#endif // MYCLASS_H
The implementation of the class and its methods go into the source file (.cpp).
Currently it includes just an empty constructor.
MyClass.cpp


More On Classes
Separate Files for Classes
XP 225
3/4
Scope Resolution Operator
The double colon in the source file (.cpp) is called the scope resolution operator, and it's used for the constructor definition:
#include "MyClass.h"
MyClass::MyClass()
{
//ctor
}
The scope resolution operator is used to define a particular class' member functions, which have already been declared. Remember that we defined the constructor prototype in the header file.
So, basically, MyClass::MyClass() refers to the MyClass() member function - or, in this case, constructor - of the MyClass class.
179 COMMENTS



Report – Report can be typed or hand written for up to two pages.

C++ programming

24/03/2020 - Wednesday.

→ More on Classes

→ Separate files Class

> Creating a New class.

* It is generally a good practice to define your new classes in separate files. This makes maintaining & reading the code easier.

* Source & header

The header file (.h) holds the function declarations (prototypes) & variable declarations. It currently includes a template for our new MyClass, with one default constructor.

* Scope Resolution Operator.

The double colon in the source file (.cpp) is called the scope resolution operator, & it's used for constructor definition.

* Source & Header.

To use our classes in our main, we need to include the header file.

> Destructors:

* Destructors are special functions, as well. They're called when an object is destroyed or deleted.

* The name of the destructor will be exactly the same as the class, only prefixed with a tilde (~).

* Since destructors can't take parameters, they also can't be overloaded.

> Selection Operator:

* #ifndef & #define.

* Member Functions

* Dot Operator

* Pointers: We can also use a pointer to access the object's members.

* Selection Operator

The arrow member selection operator (->) is used to access an object's members with a pointer.

> Const Objects

Constants

* A constant is an expression with a fixed value. It cannot be changed while the program is running.

* As with the built-in data types, we can make class objects constant by using the const keyword.

* Only non-const objects can call non-const functions.

A constant object can't call regular functions. Hence, for a constant object to work you need a constant object

> Member Initializer

- C++ provides a handy syntax for initializing members of the class called the member initializer list (also called constructor initializer)

- The member initialization list may be used for regular variables, & must be used for constant variables

> Composition part:

> Composition Part:

> The friend Keyword.

- The function someFunc() is defined as a regular function outside the class. It takes an object of type myClass as its parameter, & is able to access the private data members of the object.

> The This Keyword

Every object in C++ has access to its own address through an important pointer called the this pointer.

> Operator Overloading:

Most of the C++ built in operators can be redefined or overloaded