

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

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.

In programming, **objects** are independent units, and each has its own **identity**, just as objects in the real world do.

An apple is an object; so is a mug. Each has its unique **identity**. It's possible to have two mugs that look identical, but they are still separate, unique objects.

174 COMMENTS

What is a Class

Objects are created using **classes**, which are actually the focal point of OOP.

The class **describes** what the object will be, but is **separate** from the object itself. In other words, a class can be described as an object's **blueprint**, description, or definition. You can use the same class as a blueprint for creating multiple different objects. For example, in preparation to creating a new building, the architect creates a blueprint, which is used as a basis for actually building the structure. That same blueprint can be used to create multiple buildings.

Programming works in the same fashion. We first define a class, which becomes the blueprint for creating objects.

Each class has a **name**, and describes **attributes** and **behavior**.

In programming, the term **type** is used to refer to a class name. We're creating an object of a particular **type**.

Attributes are also referred to as **properties** or **data**

154 COMMENTS

Classes and Objects
Example of a Class

XP 219

1/3

Declaring a Class

Begin your class definition with the keyword **class**. Follow the keyword with the class name and the class body, enclosed in a set of curly braces.

The following code declares a class called **BankAccount**:

```
class BankAccount {  
};
```

A class definition must be followed by a **semicolon**.

115 COMMENTS

Q&A →

Classes and Objects
Abstraction

XP 224

1/3

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**.

A good real-world example is a *book*: When you hear the term *book*, you don't know the exact specifics, i.e., the page count, the color, the size, but you understand the **idea of a book** - the abstraction of the book.

The concept of **abstraction** is that we focus on essential qualities, rather than the specific characteristics of one particular example.

132 COMMENTS

Q&A →

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

C++ Programming

26/01/2020 - Wednesday.

Module:

→ 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 even they are completely different objects.

* In OOP, an object is self-contained, with its own identity. It is separate from other objects.

What is class

* Objects are created using classes, which are basically the template 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

* Assign your class definitions 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 very limited information to the outside world. It depends on inheritance, virtual functions without overriding implementation worldwide.

* Abstraction means; that we can have an idea of a concept that is completely separate from any specific instance.

* Encapsulation:

It's the process of the class encapsulates all the data. It is used along "visibility", not just to keep "private" inside the class, but also to protect it.

* Example of Encapsulation:

* Access Specifiers

| |
|---|
| <ul style="list-style-type: none"> * They are used to do access, hide & protect member functions. The three access levels are public, protected & private. |
| <ul style="list-style-type: none"> * A public member is accessible from outside the class, for further within the scope of the class object. |
| <ul style="list-style-type: none"> * Private: A private member cannot be accessed from outside the class, it can be accessed only from within the class. |
| <p>> Constructors:</p> <ul style="list-style-type: none"> - Class constructors are special member functions of a class. They are executable whenever objects are created within that class. The constructor's name is understood to be that of the class. It has no return type, but returns void. - Constructors can be easily used for setting initial values for certain member variables. - If default constructor does not fit, 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.

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

More On Classes
Destructors

XP 229
1/6

Destructors

Remember constructors? They're special member functions that are automatically called when an object is created. **Destructors** are special functions, as well. They're called when an object is **destroyed** or **deleted**.

Objects are destroyed when they go out of scope, or whenever the **delete** expression is applied to a pointer directed at an object of a class.

112 COMMENTS

Q&A

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

C++ Programming

2019-09-11

- New Class
 - Separate .h file
 - Creating a new class
 - It is generally a good practice to define your in the class which separate from the file maintaining readability of the code
- Source of Header
 - The header file (.h) holds the function declarations (prototypes) of variables also methods. It generally includes a template, function prototype, with some default constants.
 - Scope Resolution Operator:
The double colon in the source file (.cpp) is called the scope resolution operator, it is used for multiple definition.
 - Source of Header:
To use classes in our main, we need to include the header file.
- Destructor:
 - Destructors are special functions, as well. They're called when objects are destroyed.

Notes

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

* Other destructors can take parameters, they're often called deconstructors.

> Destructor Operator:

- friend function
- Member Functions
- D & Operator
- Pointers: We can also use a pointer to access the object members.
- Selection Operator:
The arrow member selection operator (→) is used otherwise on objects members with a pointer.

> Const Objects

• 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 regular functions. Hence, if a constant object do work you need a const object.

Member Initializers

* C++ provides a handy syntax for initializing members of the class and left the member initializers until after default constructor (initialization)

* The member initializations just may be used for member variables, but must be used first. Constructors are available.

Composition part 1

Composition part 2

The friend keyword

* The function can `friend()` is defined as a regular function outside the class. It takes an object of my class as the parameter, but with all access to private data members of the object.

The this keyword

* Every object in C++ has access to it's own address through an important pointer called the `this` pointer.

Operator Overloading

* Next to the C++ built-in operators will be introduced.