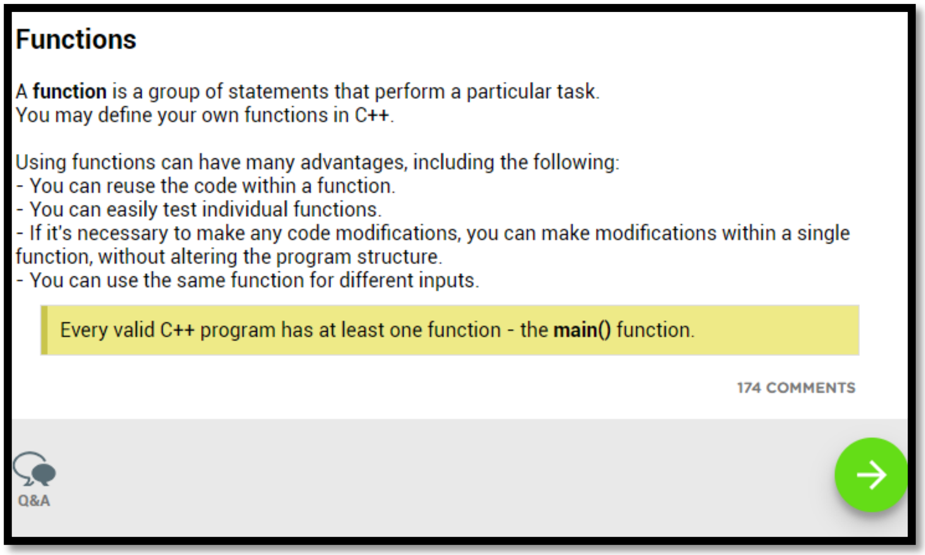
DAILY ASSESSMENT

Date:	23/06/2012	Name:	Davis S. Patel
Course:	Programming in C++	USN:	4AL16EC045
Topic:	Module 3 & 4	Semester & Section:	8th -A
GitHub Repository:	Davis		

FORENOON SESSION DETAILS

Image of session

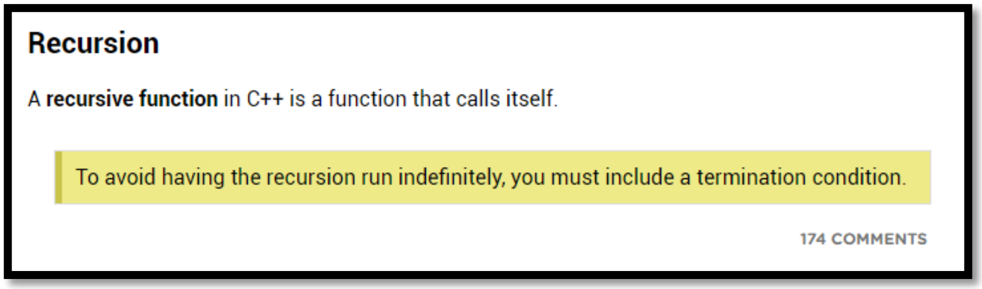


Functions

A **function** is a group of statements that perform a particular task.
You may define your own functions in C++.

Using functions can have many advantages, including the following:
- You can reuse the code within a function.
- You can easily test individual functions.
- If it's necessary to make any code modifications, you can make modifications within a single function, without altering the program structure.
- You can use the same function for different inputs.

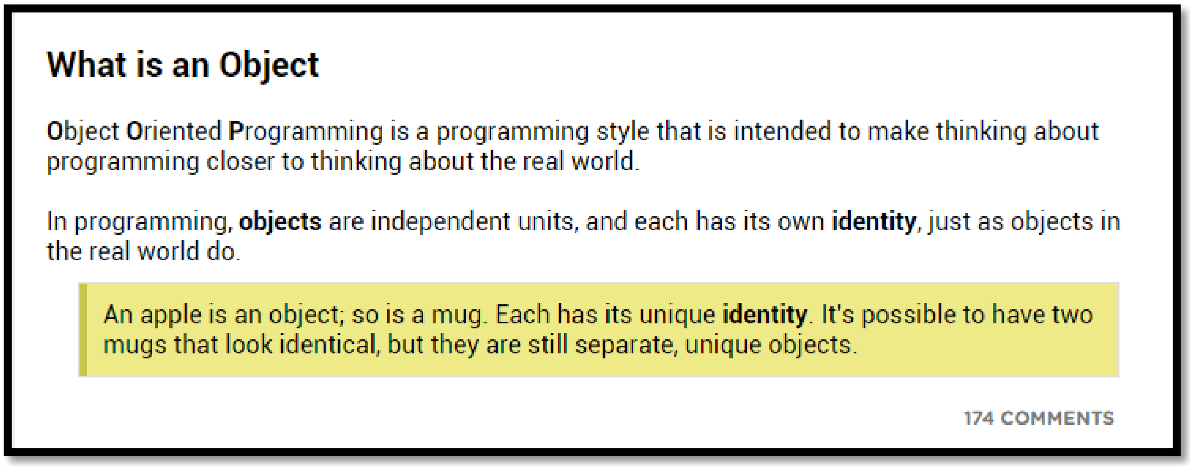
> Every valid C++ program has at least one function - the **main()** function.



Recursion

A **recursive function** in C++ is a function that calls itself.

> To avoid having the recursion run indefinitely, you must include a termination condition.



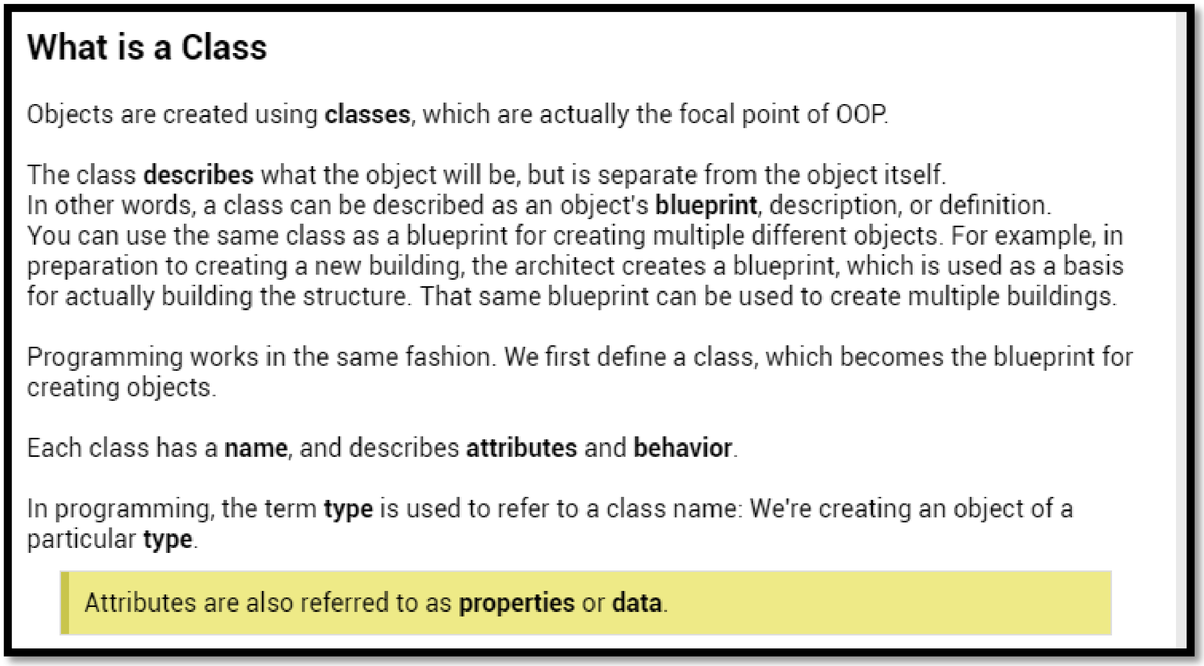
What is an Object

Object **O**riented **P**rogramming 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**.

REPORT –

C++ Functions

A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

Functions are used to perform certain actions, and they are important for reusing code: Define the code once, and use it many times.

Create a Function

C++ provides some pre-defined functions, such as main(), which is used to execute code. But you can also create your own functions to perform certain actions.

To create (often referred to as *declare*) a function, specify the name of the function.

Call a Function

Declared functions are not executed immediately. They are "saved for later use", and will be executed later, when they are called.

To call a function, write the function's name followed by two parentheses () and a semicolon;

In the following example, myFunction() is used to print a text (the action), when it is called.

Function Declaration and Definition

A C++ function consist of two parts:

- **Declaration:** the function's name, return type, and parameters (if any)
- **Definition:** the body of the function (code to be executed)

Parameters and Arguments

Information can be passed to functions as a parameter. Parameters act as variables inside the function. Parameters are specified after the function name, inside the parentheses.

Function Overloading

With **function overloading**, multiple functions can have the same name with different parameters:

Example

```
int myFunction(int x)
float myFunction(float x)
double myFunction(double x, double y)
```

C++ Classes/Objects

C++ is an object-oriented programming language.

Everything in C++ is associated with classes and objects, along with its attributes and methods. For example: in real life, a car is an object. The car has attributes, such as weight and color, and methods, such as drive and brake.

Attributes and methods are basically variables and functions that belongs to the class. These are often referred to as "class members".

A class is a user-defined data type that we can use in our program, and it works as an object constructor, or a "blueprint" for creating objects.

Create a Class

To create a class, use the `class` keyword:

Example

Create a class called "MyClass":

```
class MyClass {       // The class
  public:             // Access specifier
    int myNum;        // Attribute (int variable)
    string myString;  // Attribute (string variable)
};
```

Create an Object

In C++, an object is created from a class. We have already created the class named MyClass, so now we can use this to create objects.

To create an object of MyClass, specify the class name, followed by the object name.

To access the class attributes (myNum and myString), use the dot syntax (.) on the object.

Example

Create an object called "myObj" and access the attributes:

```
class MyClass {       // The class
  public:             // Access specifier
    int myNum;        // Attribute (int variable)
    string myString;  // Attribute (string variable)
};
```

```
int main() {
  MyClass myObj;  // Create an object of MyClass

  // Access attributes and set values
  myObj.myNum = 15;
  myObj.myString = "Some text";

  // Print attribute values
  cout << myObj.myNum << "\n";
  cout << myObj.myString;
  return 0;
}
```

Class Methods

Methods are **functions** that belongs to the class.

There are two ways to define functions that belongs to a class:

- Inside class definition
- Outside class definition