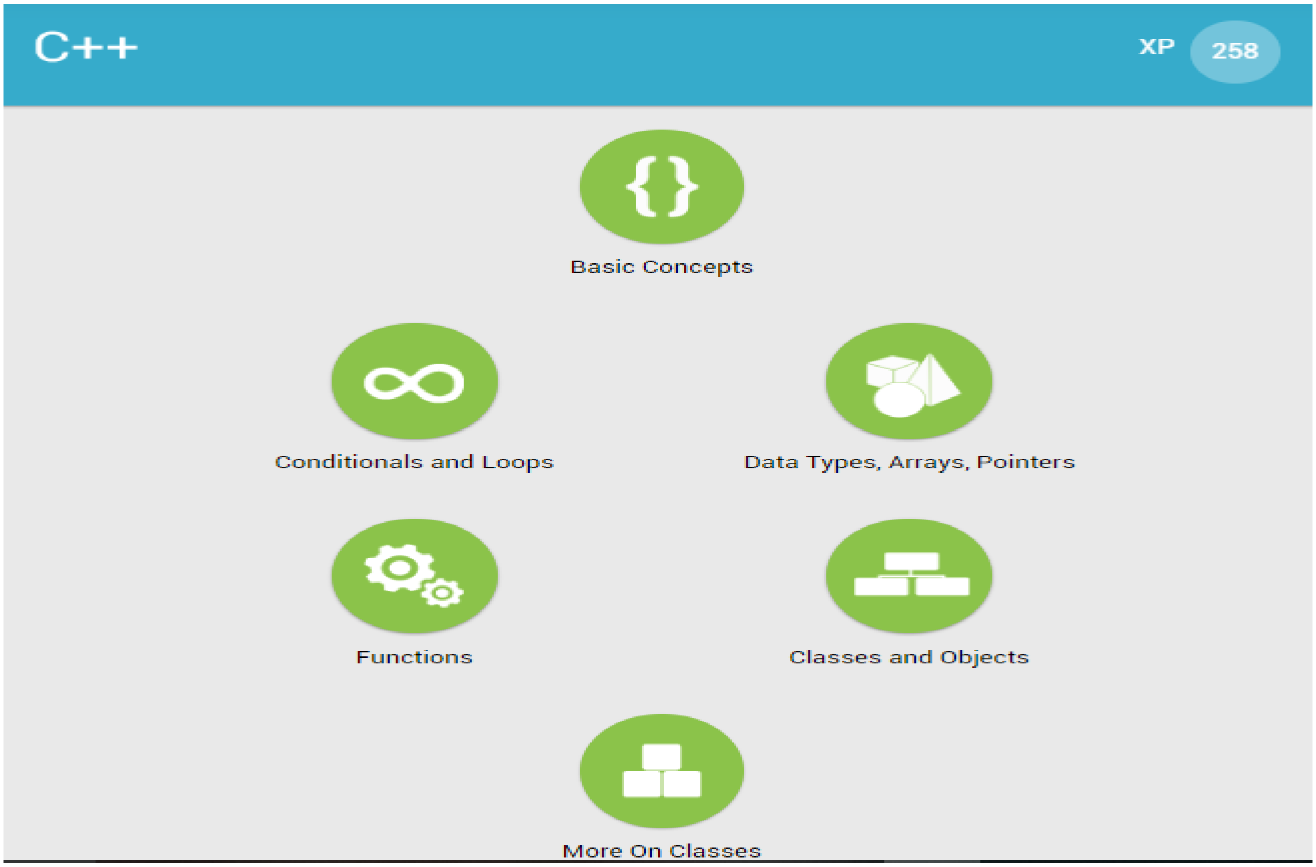
DAILY ASSESSMENT FORMAT

Date:	24/06/2020	Name:	Akshatha M Deshpande
Course:	SoloLearn	USN:	4AL17EC006
Topic:	C++ Programming	Semester & Section:	6th Sem A sec
Github Repository:	AkshathaDeshpande		

FORENOON SESSION DETAILS
Image of session  C++ XP 258 Basic Concepts Conditionals and Loops Data Types, Arrays, Pointers Functions Classes and Objects More On Classes

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

Module 5:

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 attribute describes the current state of an object.

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.
- The process of creating objects is called instantiation.
- A class definition must be followed by a semicolon.

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.

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.
- In object orientation, encapsulation means more than simply combining attributes and behavior together within a class; it also means restricting access to the inner workings of that class.

Encapsulation are:

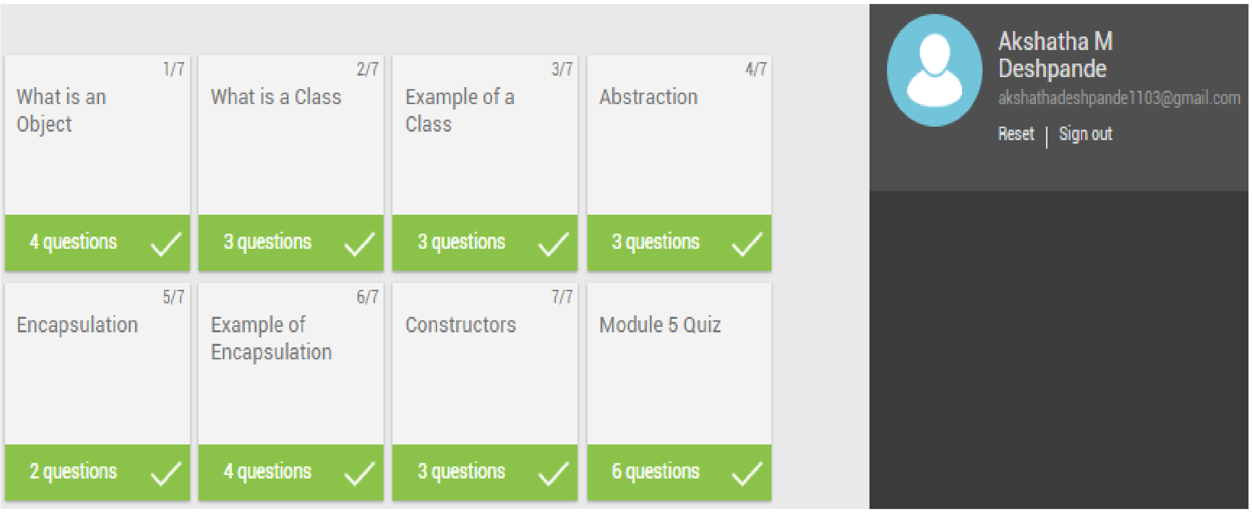
- Control the way data is accessed or modified.
- Code is more flexible and easy to change with new requirements.
- Change one part of code without affecting other part of code.

Access Specifiers:

- Access specifiers are used to set access levels to particular members of the class.
- The three levels of access specifiers are public, protected, and private.
- A public member is accessible from outside the class, and anywhere within the scope of the class object.

Constructors:

- 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 the class. It has no return type, not even void.



MODULE 6:

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.

Destructors:

- 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.

Constants:

- A constant is an expression with a fixed value. It cannot be changed while the program is running.
- Use the const keyword to define a constant variable.
- All const variables must be initialized when they're created. In the case of classes, this initialization is done via constructors.
- If a class is not initialized using a parameterized constructor, a public default constructor must be provided - if no public default constructor is provided, a compiler error will occur.

Member Initializers:

- Recall that constants are variables that cannot be changed, and that all const variables must be initialized at time of creation.
- C++ provides a handy syntax for initializing members of the class called the member initializer list (also called a constructor initializer).

Composition:

- In the real world, complex objects are typically built using smaller, simpler objects. For example, a car is assembled using a metal frame, an engine, tires, and a large number of other parts. This process is called composition.
- In C++, object composition involves using classes as member variables in other classes.
- This sample program demonstrates composition in action. It contains Person and Birthday classes, and each Person will have a Birthday object as its member.

Friend Functions:

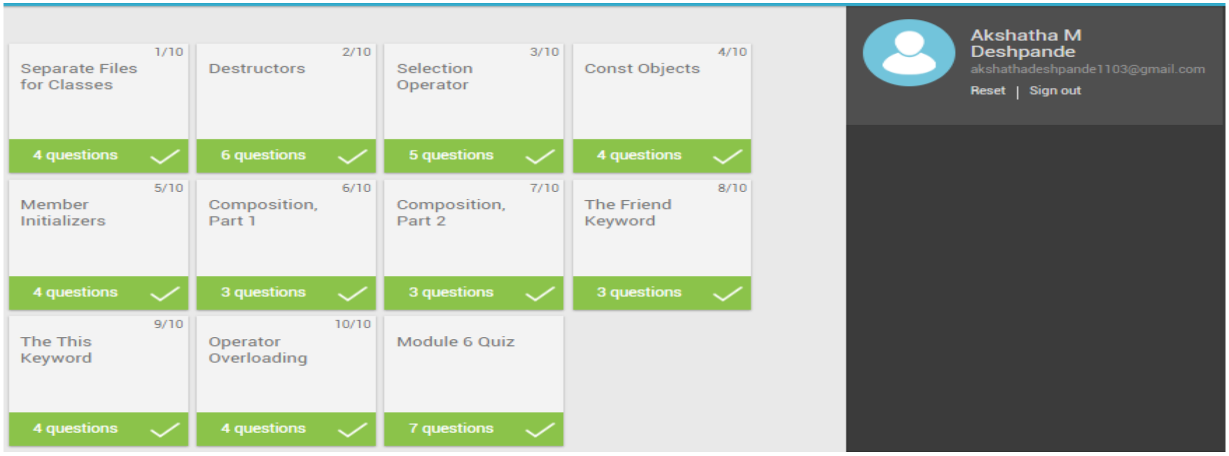
- Normally, private members of a class cannot be accessed from outside of that class.
- However, declaring a non-member function as a friend of a class allows it to access the class' private members.
- This is accomplished by including a declaration of this external function within the class, and preceding it with the keyword friend.

This:

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

Operator Overloading:

- Overloaded operators are functions, defined by the keyword operator followed by the symbol for the operator being defined.
- An overloaded operator is similar to other functions in that it has a return type and a parameter list.



1/10 Separate Files for Classes — 4 questions ✓

2/10 Destructors — 6 questions ✓

3/10 Selection Operator — 5 questions ✓

4/10 Const Objects — 4 questions ✓

5/10 Member Initializers — 4 questions ✓

6/10 Composition, Part 1 — 3 questions ✓

7/10 Composition, Part 2 — 3 questions ✓

8/10 The Friend Keyword — 3 questions ✓

9/10 The This Keyword — 4 questions ✓

10/10 Operator Overloading — 4 questions ✓

Module 6 Quiz — 7 questions ✓

Akshatha M Deshpande
akshathadeshpande1103@gmail.com