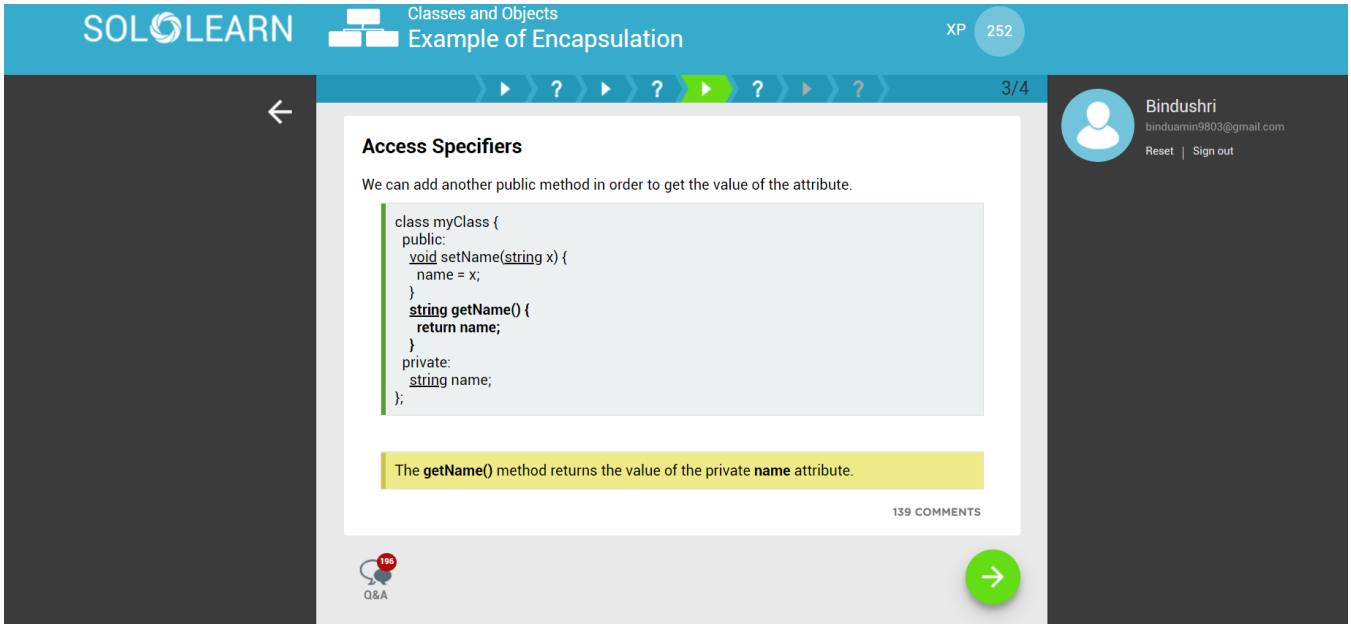
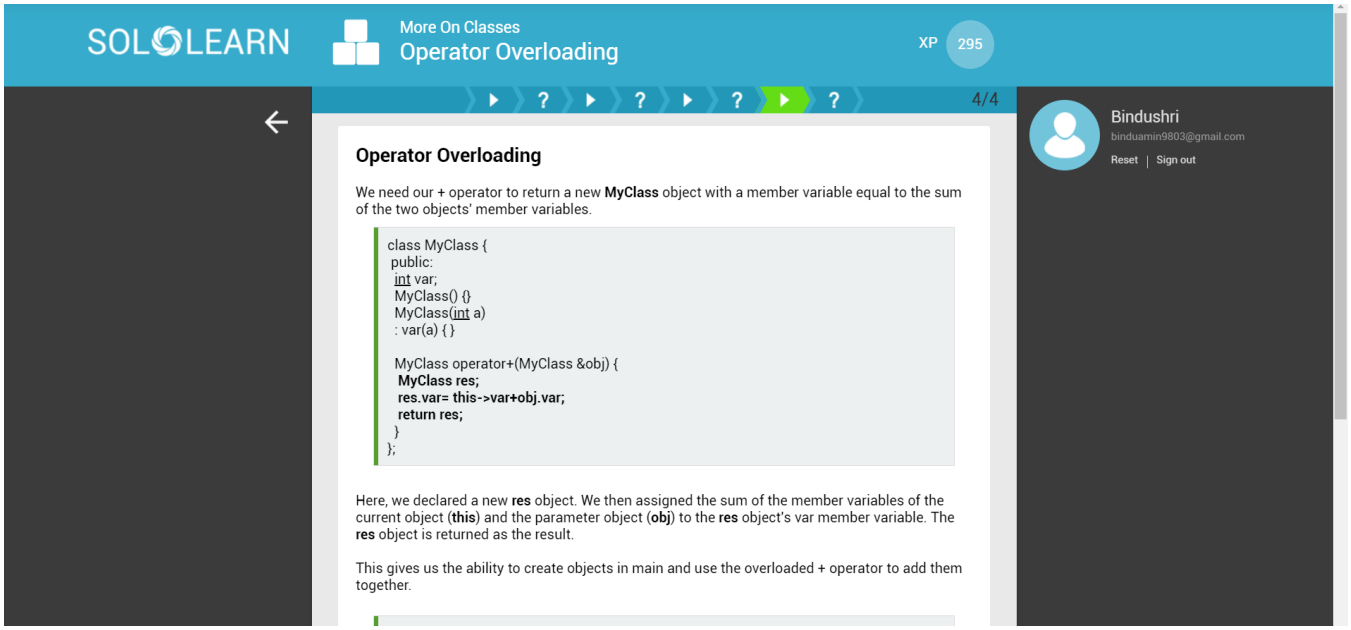
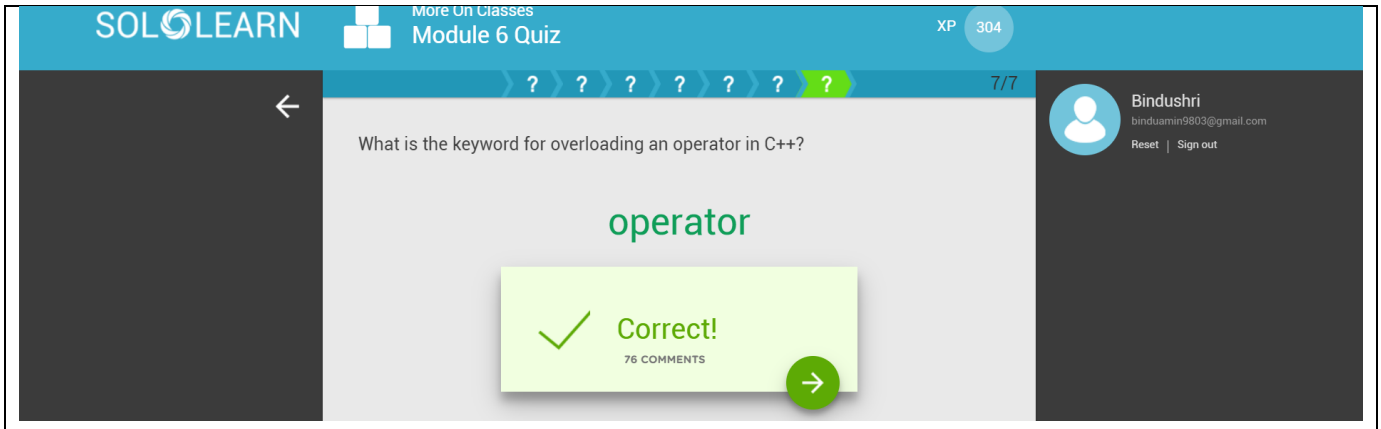
DAILY ASSESSMENT FORMAT

Date:	**24-06-2020**	**Name:**	**BINDUSHRI**
Course:	**C++ programming**	**USN:**	**4AL17EC011**
Topic:	**Classes & objects** **More on classes**		**6th A**
Github Repository:	**Bindushri**		

FORENOON SESSION DETAILS






SOLOLEARN
More On Classes
Module 6 Quiz
XP
304
7/7
What is the keyword for overloading an operator in C++?
operator
Correct!
76 COMMENTS
Bindushri
binduamin9803@gmail.com
Reset | Sign out

04/06/2020

classes & objects

OOP as a program ming style that is intended to make thinking about programming closer to the way we think about the real world.

class:- objects are created using classes, which are actually the focal point of oop.

Class describes what the object be, but is seperate from the object itself.

ex:- name: Bankaccount
attribute: account number, balance, datepened
behaviour: open(), closed(), deposit()

ex: class Bankaccount {
};

ex: Class Bankaccount {
public:- void sayHi(){
cout << "Hi" << endl;
}
};

Abstraction

Data abstraction is the concept of providing only essential information to the outside world.

Encapsulation

more than simply combining attributes and behavior together with in a class; it also means restricting access to the inner workings of that class.

Black-boxing which refers to closing the inner workings bones of the object, except of the piece that want to make public.

ex

access specifier.

```
#include<iostream>
#include<string>
using namespace std;
class myclass{
public: string name;
};
int main(){
  myclass myobj;
  myobj.name = "sololearn";
  cout << myobj.name;
  return 0;
}
```

private:- cannot be accessed or viewed.

public:- member may be used to access private.

#include <iostream>
include<string>
using namespace std;
class myclass {
public:
void setName (string x){
name=x;
}
private:
string name;
};
int main(){
myclass myobj;
myobj. setname ("John");
return 0;
}

In order to access name. from public can make use of setname() → in public!

ex.

Public:
void Setname(string x){
name =x;
}
string getName(){
return name;
}
Private:
string name;
};

getname() method returns the values of private name attribute.

Constructors

They are created & executed when ever new objects are created within that class

class myclass {
public:
myclass(){
cout << "Hey";
}void setName (string x){
name=x;
}
string getName(){
return name.
}
private:
string name;
};
int main()
{
myclass myobj;
return 0;
}

useful for setting default values

class myclass {
public:
myclass (string nm){
setName(nm);
}void setName (string x){
name=x;
}string getName(){
return name;
}private:
string name;};

int main()
{
myclass obj ("Daniel");
myclass obj ("Amy");
cout << obj. getname();
}

More on class

Creating a new class

Source & header

header file(.h) holds the function declarations (prototypes) & variable declarations

default constructor :- myclass.h

#ifndef MYCLASS_H
define MYCLASS_H

class myclass
{ public:
myclass();
protected:
private:
};

Destructor

They're called when an object is destroyed or deleted

class myclass {
public:
~Myclass(){
//some code
}
};

#include "Myclass.h"
include <iostream>
using namespace std;
Myclass :: Myclass()
{ cout << "constructor" << endl;
}
Myclass :: ~ Myclass()
{ cout << "Destructor" << endl;
}

Memberfunction

#include "Myclass.h"
include <iostream>
using namespace std;
Myclass::Myclass(){
}
void Myclass::myprint(){
cout << "Hello" << endl;
}

Pointers

Myclass obj;
Myclass* ptr = &obj;
ptr->myprint()

constant object

const int x= 42;

Completion

class person {
public:
Person (string n, Birthday b)
: name(n), bd(b)
{
private:
string name;
Birthday bd;
};

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

Random numbers

Pseudo ~~num~~random number generator function that's called rand()

when used we are required to include <cstdlib>

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main(){
  cout<<rand();
}
```

srand() function is used to generate truly random numbers.

this allows to specify a seed value as its parameter which is used for rand()

```
int main(){
srand(98);
for (int x=1; x<=10; x++){
cout<<1+(rand()%6)<<
endl;
}
}
```

this makes use of time() function to get the number of seconds on your system then randomly seed the rand()

Default values

function over loading

Recursion

A recursive function is a function that calls itself.

$4! = 4 \times 3 \times 2 \times 1 = 24$

Pass by Reference with Pointers

Function arguments:-

there are two ways to pass arguments to a function as the function being called.

By value:

By reference:

X — X — X