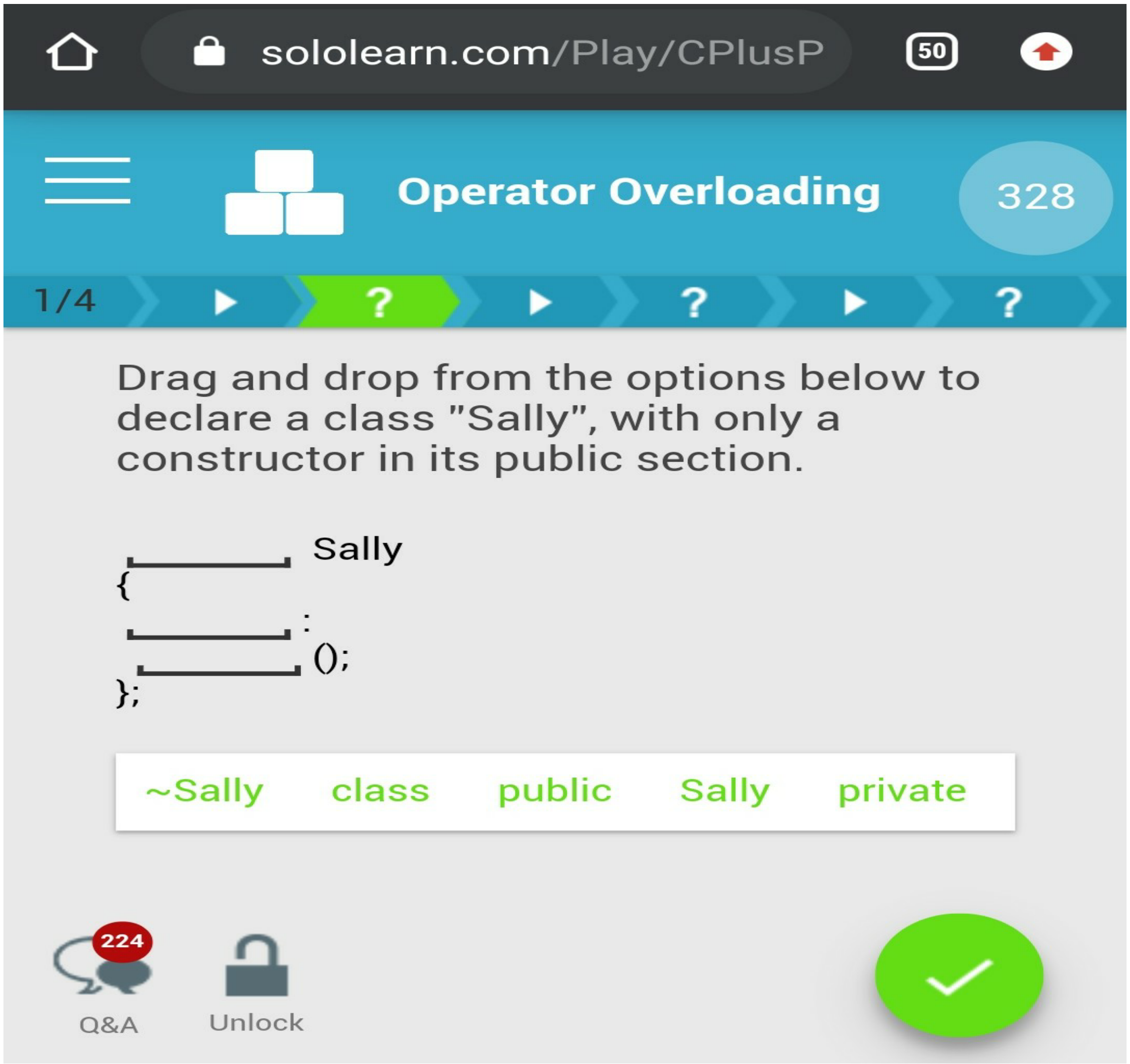
DAILY ASSESSMENT FORMAT

Date:	26-06-2020	Name:	Abhishek
Course:	C++ Programming	USN:	4al17ec001
Topic:	• Classes and Objects • More on classes	Semester & Section:	6 & 'A'
Github Repository:	Abhishek-online-courses		

SESSION DETAILS
Image of session 

Drag and drop from the options below to declare a class "Sally", with only a constructor in its public section.

```
_______ Sally
{
_______ :
  _______ ();
};
```

~Sally class public Sally private

Report –

Module 5 : classes and Objects

- Object Oriented Programming is a programming style that is intended to make thinking about programming closer to thinking about the real world.
- 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.

- **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.
- **Encapsulation**
  - ✓ 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.

- **Acces Specifier**
  - ✓ 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.

- **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 : More on Classes

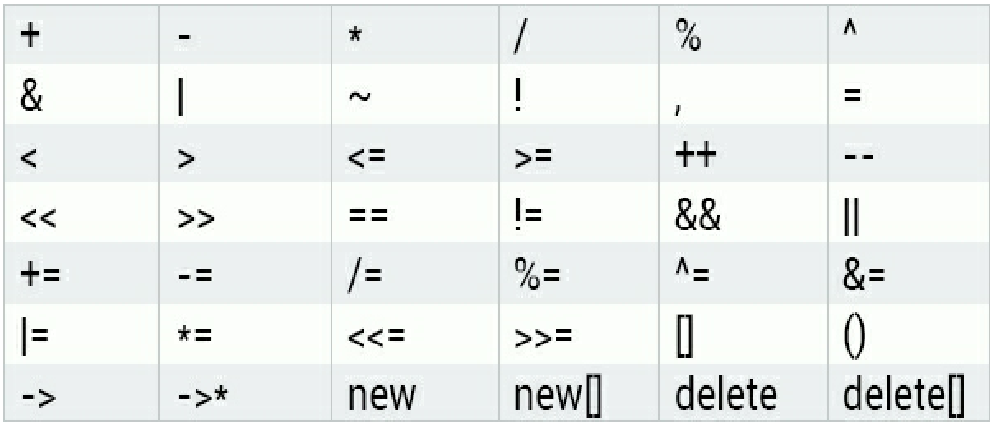
- **Destructors**
  - ✓ Destructors are special functions, as well.
  - ✓ They're called when an object is destroyed or deleted.
  - ✓ The name of a destructor will be exactly the same as the class, only prefixed with a tilde (~).
  - ✓ A destructor can't return a value or take any parameters.

- **Constant Objects**
  - ✓ As with the built-in data types, we can make class objects constant by using the const keyword.
  - ✓ When we use const to declare an object, we can't change its data members during the object's lifetime.
  - ✓ Only non-const objects can call non-const functions.

- **Friend Function**
  - ✓ 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 Pointer**
  - ✓ Every object in C++ has access to its own address through an important pointer called the this pointer.
  - ✓ Inside a member function this may be used to refer to the invoking object.

- **Operator Overloading**
  - ✓ Most of the C++ built-in operators can be redefined or overloaded.
  - ✓ Thus, operators can be used with user-defined types as well (for example, allowing you to add two objects together).
  - ✓ This chart shows the operators that can be overloaded.



+	-	*	/	%	^
&	|	~	!	,	=
<	>	<=	>=	++	--
<<	>>	==	!=	&&	||
+=	-=	/=	%=	^=	&=
|=	*=	<<=	>>=	[]	()
->	->*	new	new[]	delete	delete[]