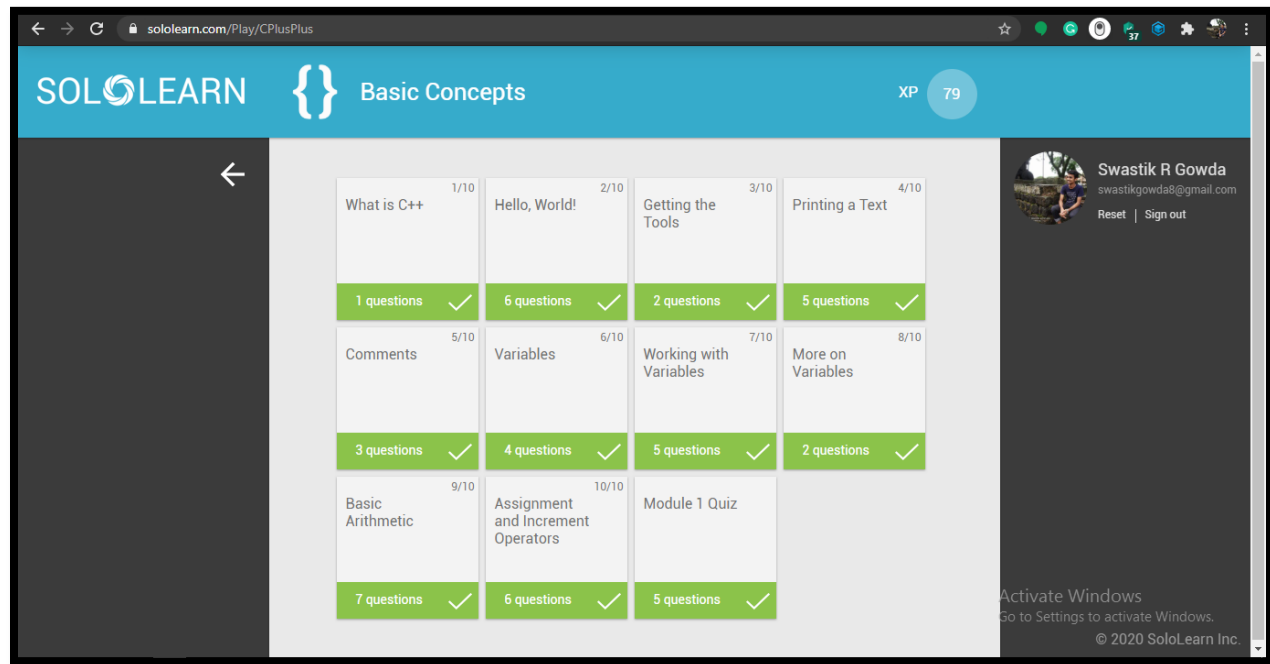


## DAILY ASSESSMENT

Date:	22-June-2020	Name:	Swastik R Gowda
Course:	Solo-Learn C++	USN:	4AL17EC091
Topic:	Module - 1 : Basic Concepts	Semester & Section:	6 <sup>th</sup> Sem 'B' Sec
Github Repository:	swastik-gowda		

### FORENOON SESSION DETAILS

#### Image of session



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

### Introduction

- ❖ C++ is a general-purpose programming language.
- ❖ C++ is used to create computer programs. Anything from art applications, music players and even video games.
- ❖ A C++ program is a collection of commands or statements.

Below is a simple code that has "Hello world!" as its output.

```
#include <iostream>
using namespace std;
int main()
{
    Cout << "Hello world!";
    return 0;
}
```

## #include <iostream>

- ❖ C++ offers various headers, each of which contains information needed for programs to work properly. This particular program calls for the header **<iostream>**.
- ❖ The number sign (#) at the beginning of a line targets the compiler's pre-processor. In this case, #include tells the pre-processor to include the **<iostream>** header.

## using namespace std;

- ❖ The C++ compiler ignores blank lines.
- ❖ In general, blank lines serve to improve the code's readability and structure.
- ❖ The line **using namespace std;** tells the compiler to use the std (standard) namespace.

## main()

- ❖ Program execution begins with the main function, **int main ()**.
- ❖ Curly brackets { } indicate the beginning and end of a function, which can also be called the function's body. The information inside the brackets indicates what the function does when executed.

## Cout

- ❖ The next line, **cout << "Hello world!";** results in the display of "Hello world!" to the screen.
- ❖ In C++, streams are used to perform input and output operations.
- ❖ In most program environments, the standard default output destination is the screen.
- ❖ In C++, cout is the stream object used to access it.
- ❖ **cout** is used in combination with the insertion operator. Write the insertion operator as << to insert the data that comes after it into the stream that comes before.

## Return

- ❖ The last instruction in the program is the **return** statement.
- ❖ The line **return 0;** terminates the **main ()** function and causes it to return the value 0 to the calling process.
- ❖ A non-zero value (usually of 1) signals abnormal termination.

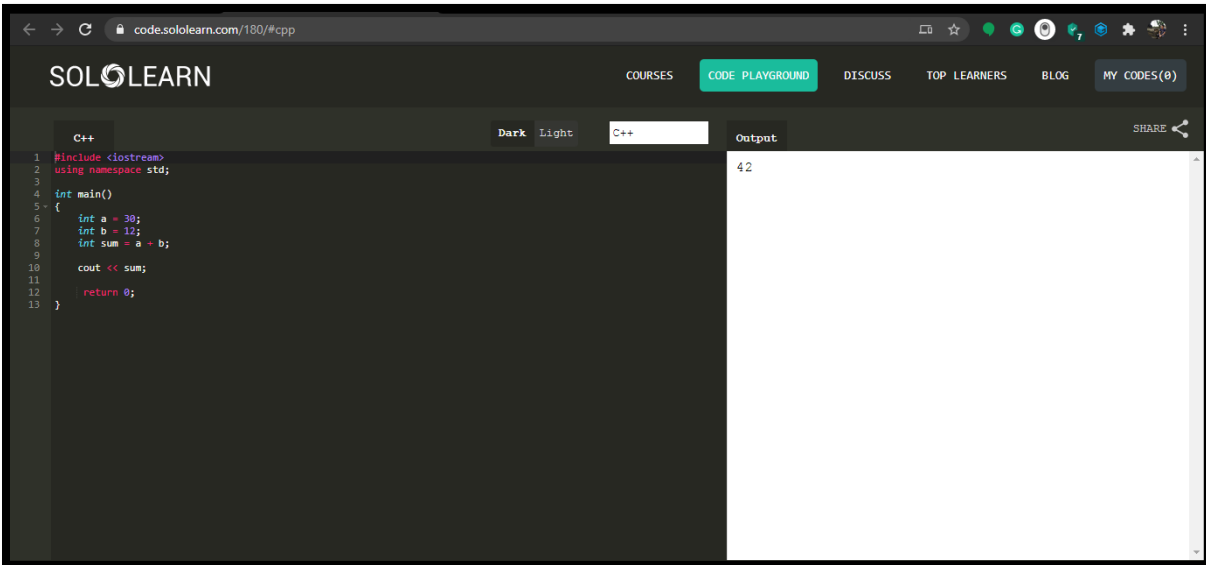
## Variables

- ❖ Creating a variable reserves a memory location, or a space in memory for storing values.
- ❖ The compiler requires that you provide a data type for each variable you declare.
- ❖ C++ offer a rich assortment of built-in as well as user defined data types.
- ❖ Integer, a built-in type, represents a whole number value. Define integer using the keyword int.
- ❖ C++ requires that you specify the type and the identifier for each variable defined.
- ❖ An **identifier** is a name for a variable, function, class, module, or any other user-defined item.
- ❖ An **identifier** starts with a letter (A-Z or a-z) or an underscore (\_), followed by additional letters, underscores, and digits (0 to 9).

```

#include <iostream>
using namespace std;
int main()
{
    int a = 30;
    int b = 12;
    int sum = a + b;
    cout << sum;
    return 0;
}

```



## Assignment Operators

- ❖ The simple assignment operator (=) assigns the right side to the left side.
- ❖ C++ provides shorthand operators that have the capability of performing an operation and an assignment at the same time.
- ❖ The same shorthand syntax applies to the multiplication, division, and modulus operators.

## Increment Operator

- ❖ The increment operator is used to increase an integer's value by one, and is a commonly used C++ operator.

The increment operator has two forms, prefix and postfix.

***++x; // prefix***

***x++; // postfix***

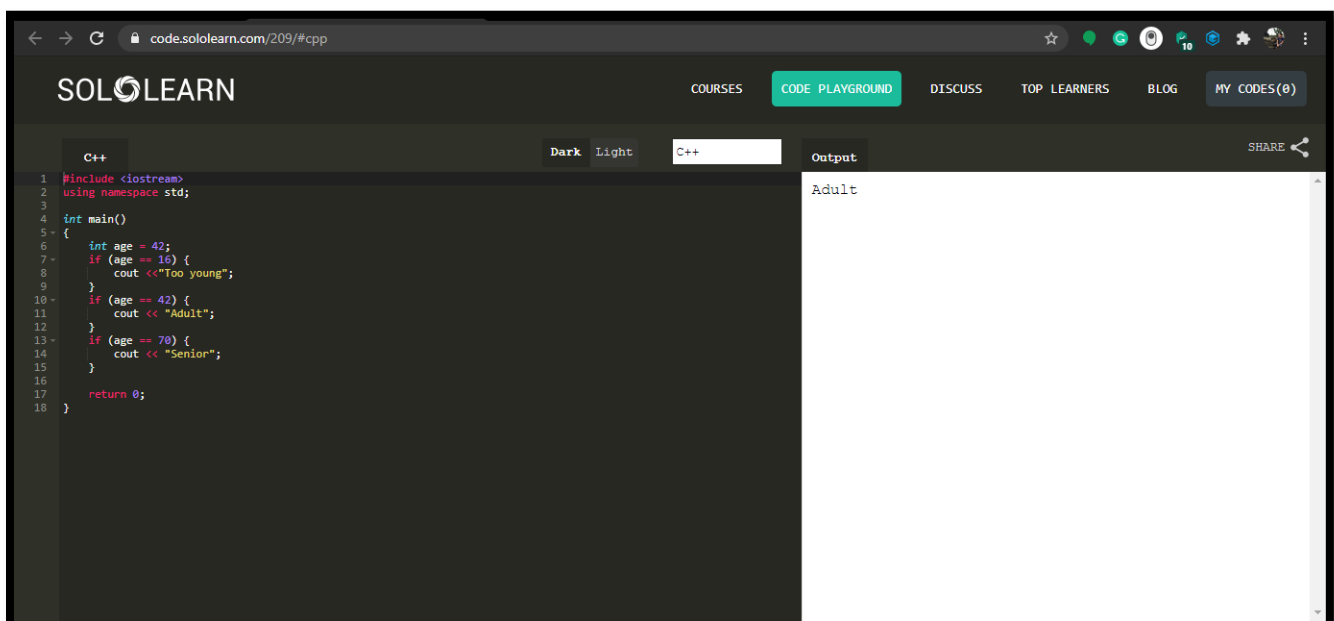
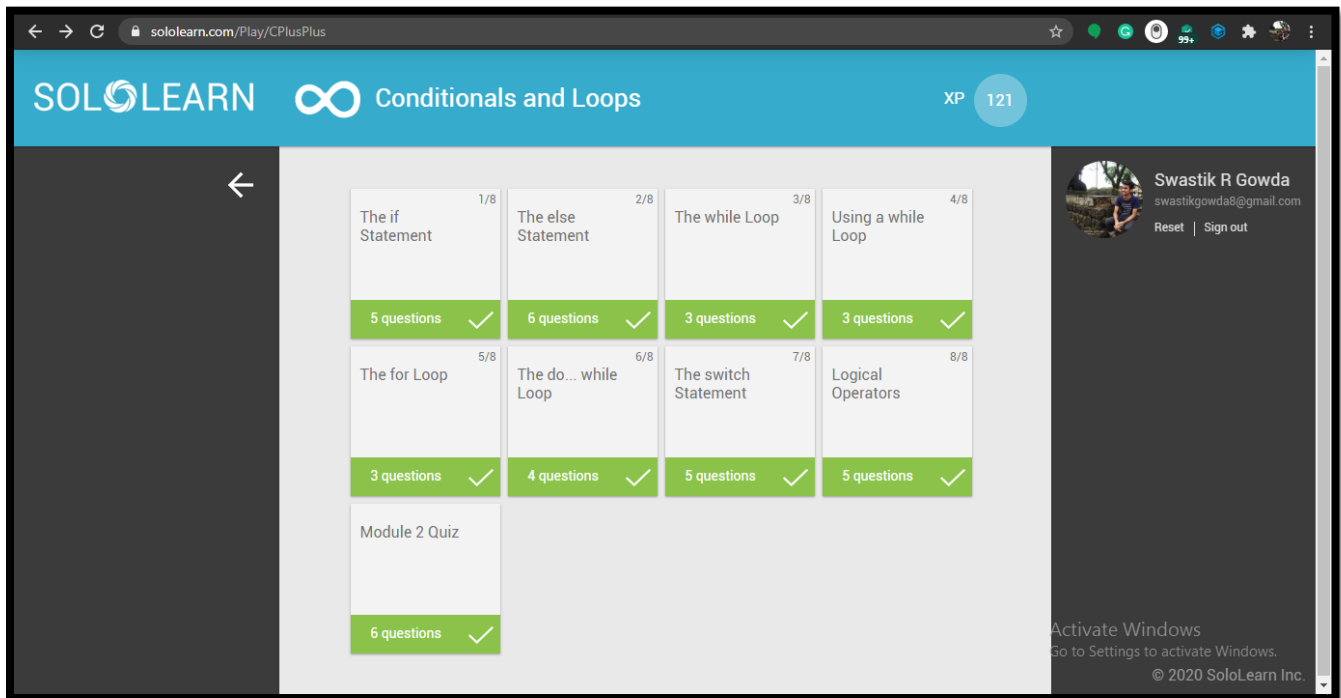
- ❖ Prefix increments the value, and then proceeds with the expression.
- ❖ Postfix evaluates the expression and then performs the incrementing.

Prefix	Postfix
<pre> x = 5; y = ++x; // x is 6, y is 6 </pre>	<pre> x = 5; y = x++; // x is 6, y is 5 </pre>

Date:	22-June-2020	Name:	Swastik R Gowda
Course:	Solo-Learn C++	USN:	4AL17EC091
Topic:	Module - 2 : Conditionals and Loops	Semester & Section:	6 <sup>th</sup> Sem 'B' Sec

## AFTERNOON SESSION DETAILS

### Image of session



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

## Conditionals

### If Statement

Use relational operators to evaluate conditions.

For example:

```
if (7 > 4)  
{  
    cout << "Yes";  
}
```

- ❖ The **if** statement evaluates the condition (7>4), finds it to be true, and then executes the cout statement.
- ❖ If we change the greater operator to a less than operator (7<4), the statement will not be executed and nothing will be printed out.

### Else Statement

- ❖ An if statement can be followed by an optional else statement, which executes when the condition is false.

#### Syntax:

```
if (condition)  
{  
    //statements  
}  
Else  
{  
    //statements  
}
```

The code above will test the condition:

- ❖ If it evaluates to true, then the code inside the if statement will be executed.
- ❖ If it evaluates to false, then the code inside the else statement will be executed.

## Loops

- ❖ A loop repeatedly executes a set of statements until a particular condition is satisfied.
- ❖ A while loop statement repeatedly executes a target statement as long as a given condition remains true.
- ❖ The loop iterates while the condition is true.

### while Loop

The loop's body is the block of statements within curly braces.

For example:

```
int num = 1;  
while (num < 6) {  
    cout << "Number: " << num << endl;  
    num = num + 1;  
}
```

Outputs
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5

The example above declares a variable equal to 1 (**int num = 1**).

The while loop checks the condition (**num < 6**), and executes the statements in its body, which increment the value of **num** by one each time the loop runs.

### Do...While Loop

- ❖ Unlike for and while loops, which test the loop condition at the top of the loop, the do...while loop checks its condition at the bottom of the loop.
- ❖ A do...while loop is similar to a while loop. The one difference is that the do...while loop is guaranteed to execute at least one time.

Syntax:

```
do {  
    statement(s);  
} while (condition);
```

### For Loop

- ❖ A for loop is a repetition control structure that allows you to efficiently write a loop that executes a specific number of times.

Syntax:

```
for ( init; condition; increment )  
{  
    statement(s);  
}
```

- ❖ The **init** step is executed first, and does not repeat.
- ❖ Next, the **condition** is evaluated, and the body of the loop is executed if the condition is true.
- ❖ In the next step, the **increment** statement updates the loop control variable.
- ❖ Then, the loop's body repeats itself, only stopping when the condition becomes false.

## Switch Statement

The switch statement tests a variable against a list of values, which are called cases, to determine whether it is equal to any of them.

```
switch (expression) {
  case value1:
    statement(s);
    break;
  case value2:
    statement(s);
    break;
  ...
  case valueN:
    statement(s);
    break;
}
```

Switch evaluates the expression to determine whether it's equal to the value in the case statement. If a match is found, it executes the statements in that case.

## Logical Operators

Operator	Name of Operator	Form
&&	AND Operator	y && y
	OR Operator	x    y
!	NOT Operator	! x

Left Operand	Right Operand	Result
false	false	false
false	true	false
true	false	false
true	true	true

Date:	22-June-2020	Name:	Swastik R Gowda
Course:	Webinar	USN:	4AL17EC091
Topic:	Trend in IT Domain	Semester & Section:	6 <sup>th</sup> Sem 'B' Sec

### AFTERNOON SESSION DETAILS

#### Image of session

The screenshot shows a Facebook Live interface for a webinar titled "Intelligence System". The main content area displays two slides. The left slide, titled "What is Intelligence System?", features a Venn diagram with three overlapping circles labeled "System", "Intelligence", and "Intelligence System". Below the diagram, it states: "An intelligent system is a machine with an embedded, Internet-connected computer that has the capacity to gather and analyze data and communicate with other systems." The right slide, titled "What is a System?", defines a system as "a collection of connected elements or components that are organized for a common purpose". Below this, another slide titled "What is Intelligence?" defines intelligence as "The ability of a system to calculate, reason, perceive relationships and analogies, learn from experience, store and retrieve information from memory, solve problems, comprehend complex ideas, use natural language fluently, classify, generalize, and adapt new situations." This slide includes a diagram titled "What is Intelligence Composed of?" showing "Intelligence" at the center, connected to five components: Reasoning, Linguistic Intelligence, Learning, Perception, and Problem Solving. The bottom of the screen shows a navigation bar with icons for Home, a lightbulb (selected), a calendar, a target, a cloud, a checklist, and a folder. There are also controls for Unmute, Start Video, Participants (118), Chat, Share Screen, Record, and a red "Leave" button.

The screenshot shows a Zoom video player interface. The main video area displays a quote by Mark Zuckerberg, Facebook CEO: "The biggest thing that we're focused on with AI is building computer services that have better perception than people". The quote is overlaid on a video of Mark Zuckerberg. The Zoom interface includes a top bar with a speaker icon, a green checkmark, the word "Zoom", and a "Leave" button. Below the video, there's a navigation bar with icons for Unmute, Home, a lightbulb, a calendar (selected), a target, a cloud, a checklist, and a folder. There are also controls for Start Video, Share, Participants, and a "More" button.



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

