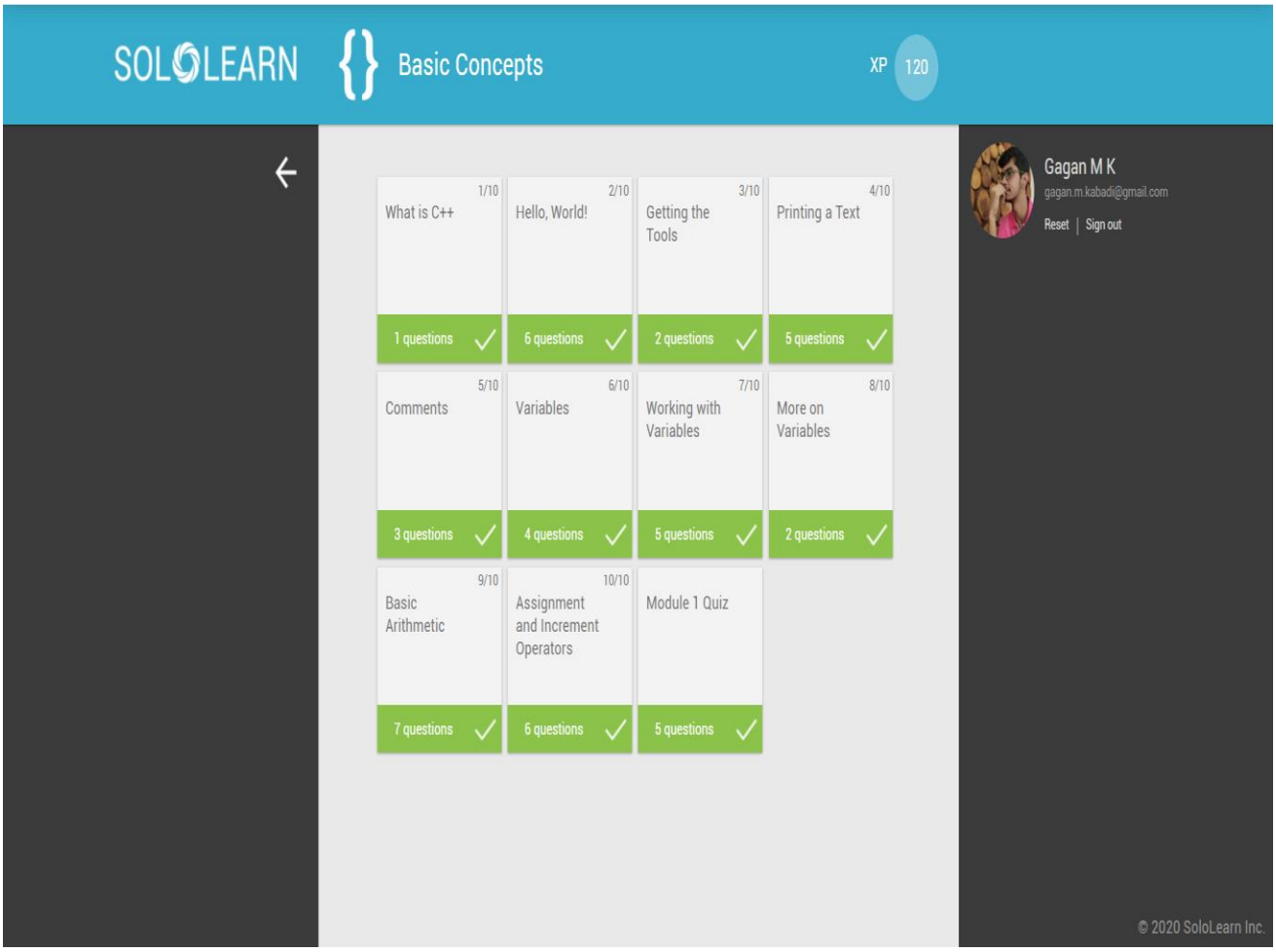
DAILY ASSESSMENT REPORT

Date:	**22 June 2020**	**Name:**	**Gagan M K**
Course:	**C Plus Plus**	**USN:**	**4AL17EC032**
Topic:	• **Basic Concepts** • **Conditionals and loops**	**Semester & Section:**	**6th sem & 'A' sec**
GitHub Repository:	**Alvas-education-foundation/Gagan-Git**		

FORENOON SESSION DETAILS

Image of session



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

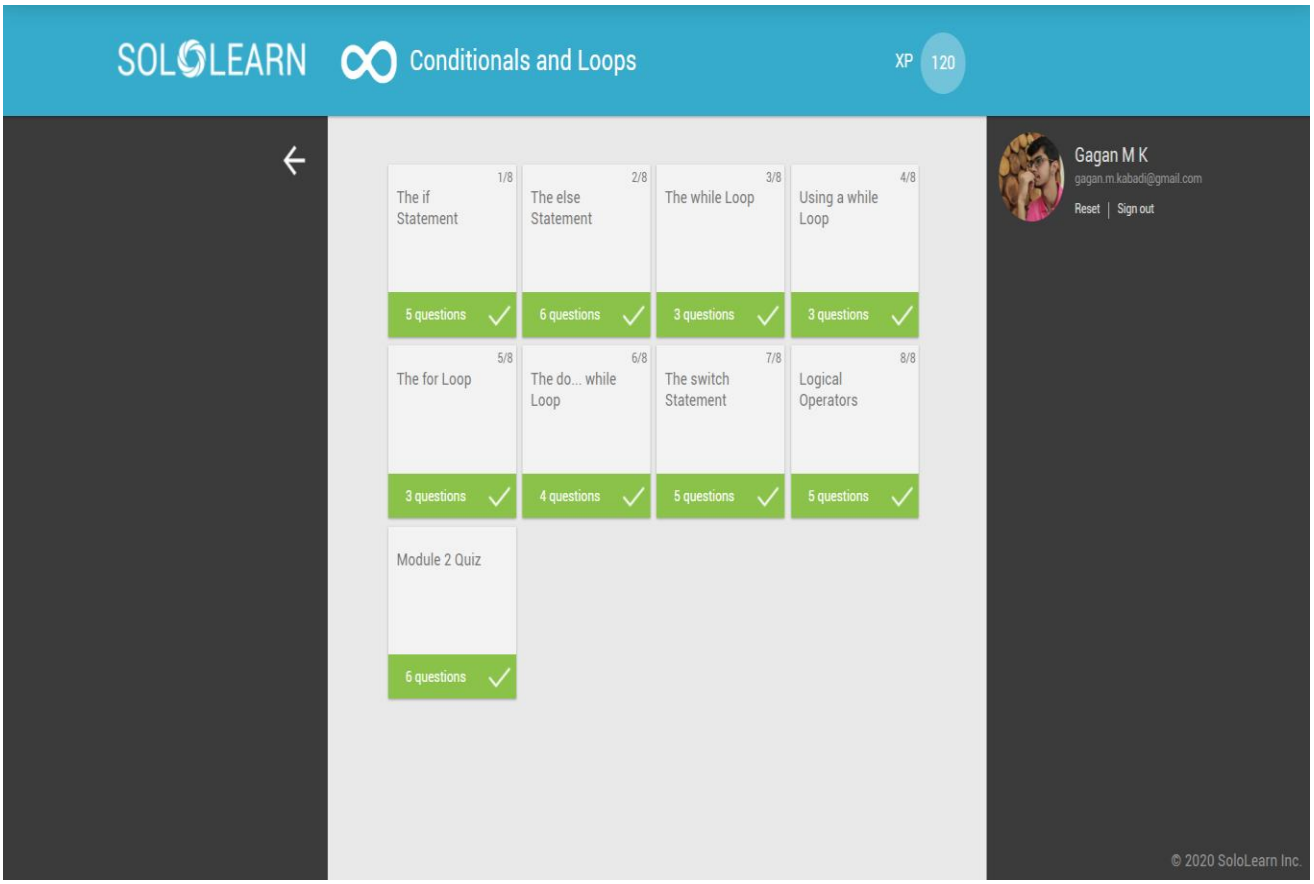
Basic Concepts:

- **C++ is a general-purpose programming language.**
- **C++ is used to create computer programs. Anything from art applications, music players and even video games!**
- **C++ was derived from C, and is largely based on it.**
- **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 a single cout statement with as many instances of \n as your program requires will print out multiple lines of text.**
- **Comments are explanatory statements that you can include in the C++ code to explain what the code is doing.**
- **The compiler ignores everything that appears in the comment, so none of that information shows in the result.**
- **A comment beginning with two slashes (//) is called a single-line comment. The slashes tell the compiler to ignore everything that follows, until the end of the line.**
- **Comments that require multiple lines begin with /* and end with */**
- **You can place them on the same line or insert one or more lines between them.**
- **Comments can be written anywhere, and can be repeated any number of times throughout the code.**
- **Within a comment marked with /* and */, // characters have no special meaning, and vice versa. This allows you to "nest" one comment type within the other.**
- **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).**
- **Define all variables with a name and a data type before using them in a program. In cases in which you have multiple variables of the same type, it's possible to define them in one declaration, separating them with commas.**
- **A variable can be assigned a value, and can be used to perform operations.**
- **For example, we can create an additional variable called sum, and add two variables together.**
- **You have the option to assign a value to the variable at the time you declare the variable or to declare it and assign a value later.**

- **C++ supports these arithmetic operator.**

Operator	Symbol	Form
Addition	+	x + y
Subtraction	-	x - y
Multiplication	*	x * y
Division	/	x / y
Modulus	%	x % y

Conditionals and loops:



- **The if statement is used to execute some code if a condition is true.**

Syntax:

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

- **The condition specifies which expression is to be evaluated. If the condition is true, the statements in the curly brackets are executed.**
- **If the condition is false, the statements are simply ignored, and the program continues to run after the if statements body.**
- **Relational Operators:**

Operator	Description	Example
>=	Greater than or equal to	7 >= 4 True
<=	Less than or equal to	7 <= 4 False
==	Equal to	7 == 4 False
!=	Not equal to	7 != 4 True

- **The not equal to operator evaluates the operands, determines whether or not they are equal. If the operands are not equal, the condition is evaluated to true.**
- **An if statement can be followed by an optional else statement, which executes when the condition is false.**

Syntax:

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

- **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.**
- **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.**
- **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);
```

- **Sometimes there is a need to test a variable for equality against multiple values. That can be achieved using multiple if statements.**
- **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**
- **Logical Operators:**

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

- **The OR (||) operator returns true if any one of its operands is true.**
- **The logical NOT (!) operator works with just a single operand, reversing its logical state. Thus, if a condition is true, the NOT operator makes it false, and vice versa.**
- **Example program:**

```
int age = 10;
if ( !(age > 16) ) {
  cout << "Your age is less than 16" << endl;
}

// Outputs "Your age is less than 16"
```

“Attended Webinar on Trend in IT Domain Conducted by Alva's Education Foundation”