Daily Assessment format

Date: 29/06/2020
Course: C++(sololearn)
Topic: Module1: Basic concepts
Module 1
Github repository: jyoti-courses

Name: Jyoti.S.Donnur
Usn: 4AL17EC037

Forenoon session details

Report

C++:

C++ is a general purpose programming language and which is used to create computer programs. Anything from art applications, music players & 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;
}
```

Main

program execution begins with the main fu^n int main

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

getting the tools

1. Integrated development environment: provides tools for writing source code. Any text editor can be used as an IDE.

2 Compiler: compiles source code into the final executable program. there are a no of c++ compilers available. the most frequently used & free available compiler is the GNU c/c++ compiler.

New Line

→ The Cout operator doesn't insert a line break at the end of the output. one way to print two lines is to use the endl manipulator. which will put in a line break.

```
#include <iostream>
using namespace std;
int main()
{
 cout<<"Hello World!"<<endl;
 cout<<"I love programming!";
 return 0;
}
```

Comments

→ 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 inform^n shows in the result.

for ex:

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

Variables

→ Creating a variable reserves a memory location, or 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 no value define integer using the keyword int.

Arithmetic operators

operator	symbol
addition	+
subtraction	−
multiplication	*
division	/
modulus	%

Assignment operators

→ The simple assignment operator(=) assigns the right side to the leftside. c++ provides shorthand operators that have the capability of performing an operation & an assignment at the same time

```
ex: int x =10;
  x+=4;// equivalent to x =x+4
  x-=5;// equivalent to x = x-5.
```

Date: 22/06/2020

Course: c++(sololearn)

Topic: Module 2: conditionals & loop

Github Repository: jyoti-course

Name: Jyoti S Donnur

Usn: 4AL17EC037

Afternoon session details

Report

conditionals & loops

decision making

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 evaluat if the condition is true, the statements in the curly brackets are executed.

The if statement

Use relational operators to evaluate conditions

```
ex: if (7>4){
      cout<<"yes";
    }
//outputs "yes"
```

The else statements

→ 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 while loop:

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

ex:

```
int num=1;
while(num<6){
  cout<<"number:"<<num<<endl;
   num=num+1;
 }
/*outputs
 number:1
 number:2
 number:3
 number:4
 number:5*/
```

For loop

A for loop is a repetition control structure that allows you to effeciently write a loop that executes a specific no of times

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

dowhile loop

Unlike for & while loops, which test the loop condition at the top of the loop, the dowhile loop checks its condition at the bottom of the loop.

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

switch statement

→ tests a variable against a list of values, which are called cases, to determine wheather it is equal to any of them.

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

Logical operators

operator	name of operator.
&&	AND operator
||	OR operator
!	NOT operator