DAILY ASSESSMENT FORMAT

Date:	**22-06-2020**	**Name:**	poorvi
Course:	**C++ programming**	**USN:**	4AL17EC071
Topic:	**Basic concepts** **Condition and loops**		6^{th} b
Github Repository:	Poorvi-2000		

FORENOON SESSION DETAILS

Fill in the blanks to declare a function printNumber and overload it. Declare one printNumber, taking an integer parameter; and another printNumber, taking a float parameter.

```
void printNumber(int x) {
 cout << "integer is " << x << endl;
}
void printNumber(float x) {
 cout << "float is " << x << endl;
}
```

Function Overloading

You **can not** overload function declarations that differ only by **return** type.
The following declaration results in an error.

```cpp
int printName(int a) { }
float printName(int b) { }
double printName(int c) { }
```

TRY IT YOURSELF

Although each function uses the same name, the only difference from one to the other is the **return** type, which is not allowed.

C++ 02 June 2020

C++ is a general-purpose programming language.

```
#include <iostream>
using namespace std;

int main()
{
  cout << "Hello world";
  return 0;
}
```

→ You can add multiple insertion operators after cout:

```
cout << "This" << "is" << "so" << "
    awesome!";
```

- cout operator doesn't insert a line break at the end of the output ('\n')

```
#include <iostream>
using namespace std;
int main()
{
  cout << "Hello world!" << endl; (or '\n')
  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.

(//) (/* */)

Variables

Creating a variable reserves a memory location, or a space in memory for storing values

Basic Arithmetic

addition +
subtraction -
multiplication *
division /
modulus %
Assignment & increment operator

x += 4
x -= 5

Conditional and loops

if statement

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

Relational operators

>= , <= , == , !=

else statement

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

Nested if statements.

```
int mark = 100;
if (mark >= 50) {
  cout << "you passed." << endl;
  if (mark == 100) {
    cout << "perfect!" << endl;
  }
}
else {
  cout << "you failed" << endl;
}
```

Nested if-else statements

```
if ( ) {
  if ( ) {
    cout << " ";
  } else { cout << " "; } }
else { if ( ) { } else { } }
```

while loop

Loop repeatedly executed a set of statements until a particular condition is satisfied.

using increment or decrement

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

for loop

```
for (int ; condition ; x++) {
  //same code
}
```

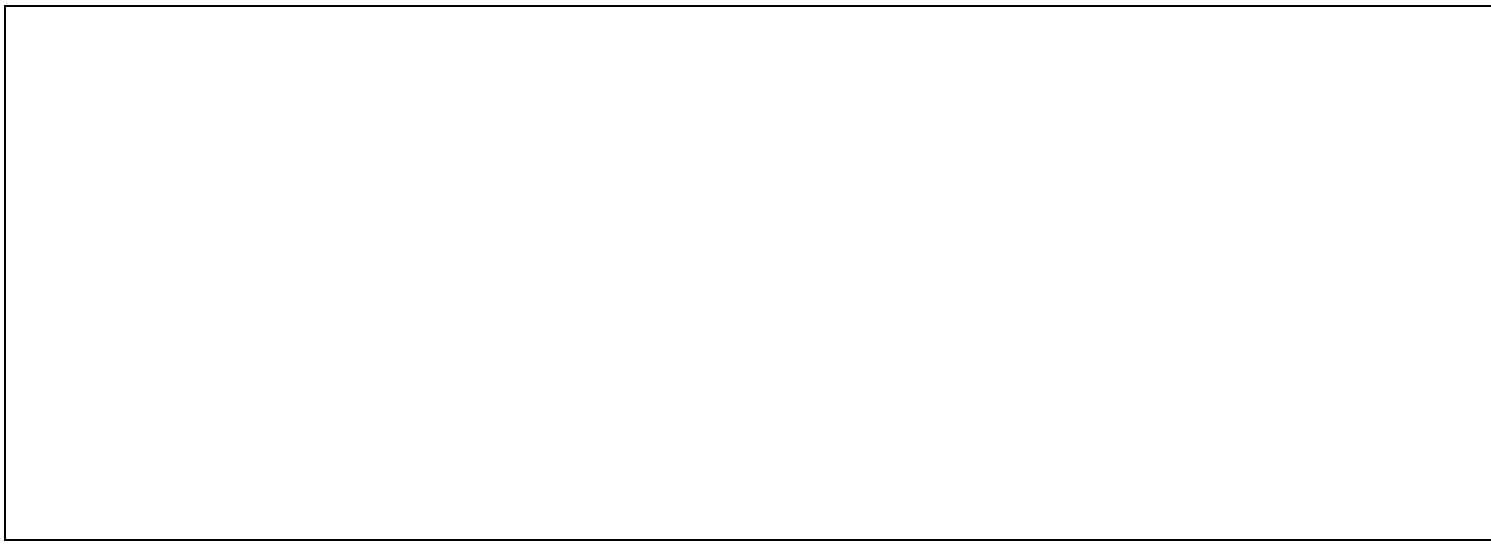
Do-while loop

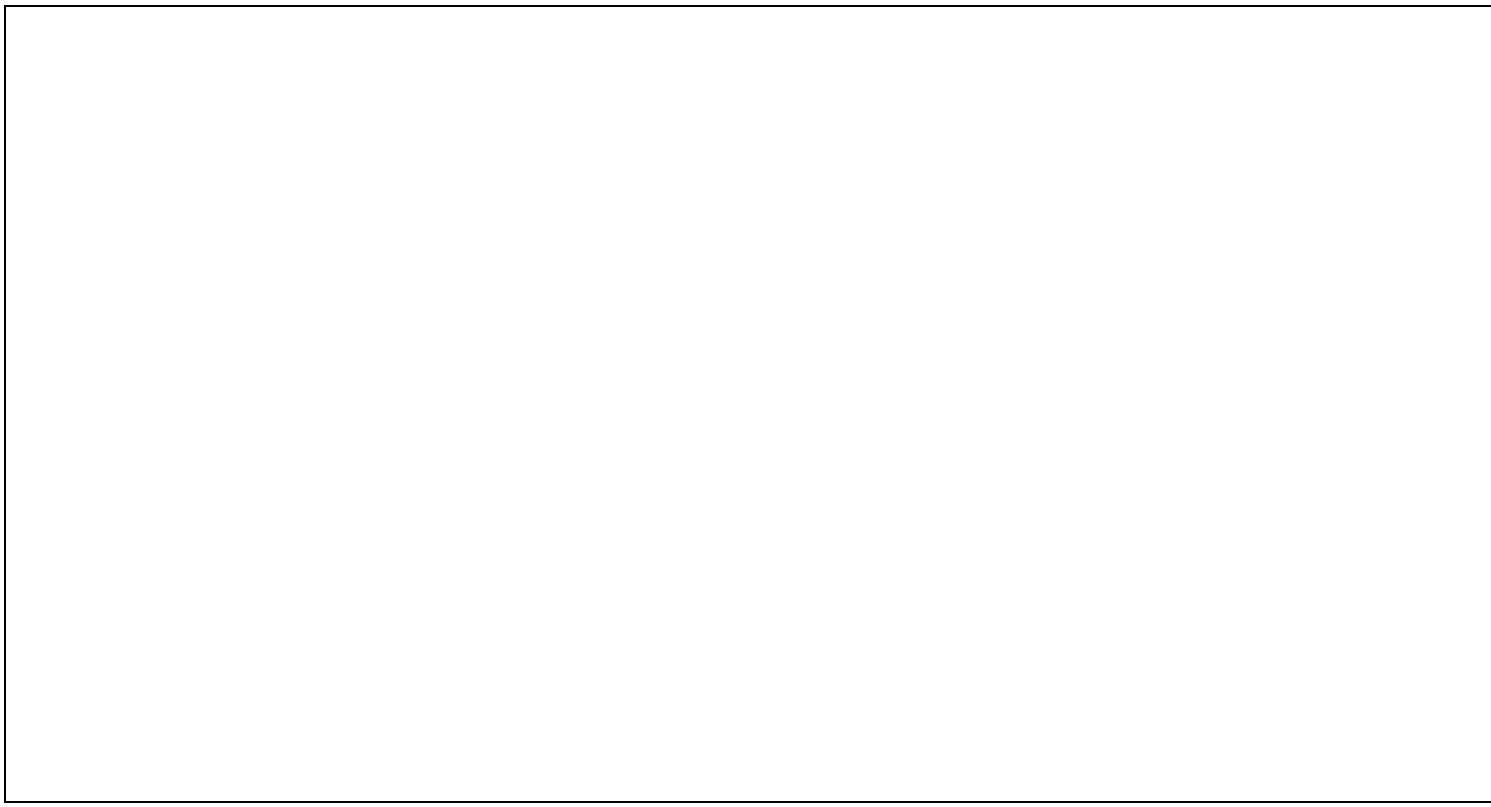
```
do {
  statement(s);
} while (condition);
```

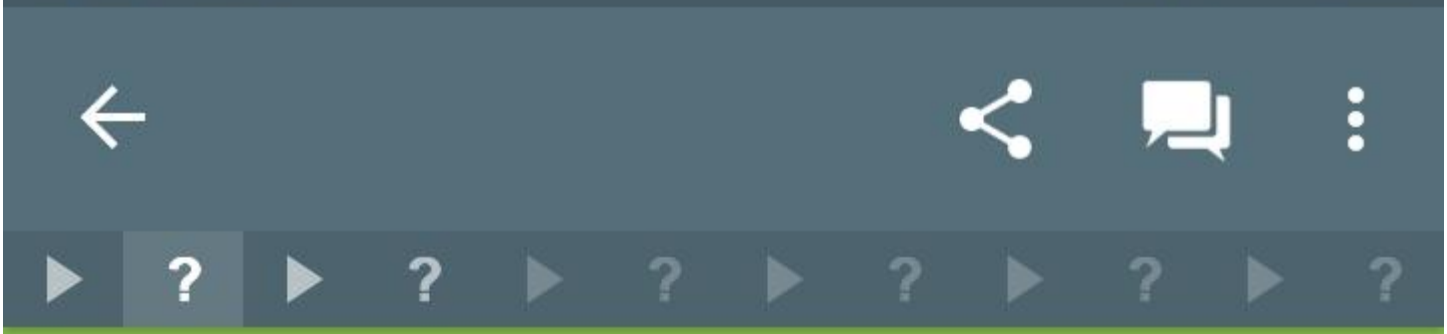
Multiple conditions

Switch statements

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







Every C++ program starts with the function:

- ○ #include
- ○ int()
- ◉ main
- ○ my_function

Basic Concepts

Control Structures

Functions & Modules

Exceptions & Files

0/9