C Programming

Write the following C code and save the file as

myfirstprogram.c

```c
#include <stdio.h>

int main() {
  printf("Hello World!");
  return 0;
}
```

Output:-

```
Hello World!
```

Note:-

- `#include <stdio.h>` is a **header file library** that lets us work with input and output functions, such as `printf()` (used in line 4). Header files add functionality to C programs.
- Another thing that always appear in a C program is `main()`. This is called a **function**. Any code inside its curly brackets `{}` will be executed.

C++ Programming

Write the following C++ code and save the file as *myfirstprogram.cpp*

Code:

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

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

output:

```
Hello World!
```

Note:-

- `#include <iostream>` is a **header file library** that lets us work with input and output objects, such as `cout` (used in line 5). Header files add functionality to C++ programs.
- `using namespace std` means that we can use names for objects and variables from the standard library.

Output (Print Text)

To output values or print text in C, you can use the `printf()` function:

Example

```
#include <stdio.h>

int main() {
  printf("Hello World!");
  return 0;
}
```

C++ Output (Print Text)

The `cout` object, together with the `<<` operator, is used to output values and print text.

Just remember to surround the text with double quotes (""):

Example

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

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

c variables

Variables are containers for storing data values, like numbers and characters.

In C, there are different **types** of variables (defined with different keywords), for example:

- `int` - stores integers (whole numbers), without decimals, such as `123` or `-123`
- `float` - stores floating point numbers, with decimals, such as `19.99` or `-19.99`
- `char` - stores single characters, such as `'a'` or `'B'`. Characters

C++ Variables

Variables are containers for storing data values.

In C++, there are different **types** of variables (defined with different keywords), for example:

- `int` - stores integers (whole numbers), without decimals, such as 123 or -123
- `double` - stores floating point numbers, with decimals, such as 19.99 or -19.99
- `char` - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
- `string` - stores text, such as

are surrounded by **single quotes**

Format Specifiers:-

Format specifiers are used together with the `printf()` function to tell the compiler what type of data the variable is storing. It is basically a **placeholder** for the variable value.

A format specifier starts with a percentage sign `%`, followed by a character.

For example, to output the value of an `int` variable, use the format specifier `%d` surrounded by double quotes (`""`), inside the `printf()` function:

Example

```
int myNum = 15;
printf("%d", myNum);

// Outputs 15
```

To print other types, use `%c` for `char` and `%f` for `float`:

Example:-

#include <stdio.h>

int main() {

```
// Create variables
int myNum = 15;              //
```

"Hello World". String values are surrounded by double quotes

- `bool` - stores values with two states: true or false

C++ Variables

Variables are containers for storing data values.

In C++, there are different **types** of variables (defined with different keywords), for example:

- `int` - stores integers (whole numbers), without decimals, such as 123 or -123
- `double` - stores floating point numbers, with decimals, such as 19.99 or -19.99
- `char` - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
- `string` - stores text, such as "Hello World". String values are surrounded by double quotes
- `bool` - stores values with two states: true or false

Example:-

Create a variable called **myNum** of type `int` and assign it the value **15**:

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

int main() {
  int myNum;
  myNum = 15;
  cout << myNum;
  return 0;
```

```
Integer (whole number)
float myFloatNum = 5.99;    //
Floating point number
char myLetter = 'D';        //
Character

// Print variables
printf("%d\n", myNum);
printf("%f\n", myFloatNum);
printf("%c\n", myLetter);
return 0;
}
```

Output:-

```
15
5.990000
D
```

String Example:-

```
char greetings[] = "Hello World!";
printf("%s", greetings);
output:-
Hello World!

Example 2:-

 #include <stdio.h>

int main() {
  int myNum = 15;
  char myLetter = 'D';
  printf("My number is %d and my
letter is %c", myNum, myLetter);
  return 0;
}
```

Output:-

```
My number is 15 and my letter is D
}
```

Output:-

```
15
```

Other Types

A demonstration of other data types:

Example

```
int myNum = 5;                  //
Integer (whole number without
decimals)
double myFloatNum = 5.99;     //
Floating point number (with
decimals)
char myLetter = 'D';          //
Character
string myText = "Hello";      //
String (text)
bool myBoolean = true;        //
Boolean (true or false)

Example 2:-

#include <iostream>
using namespace std;

int main() {
  int myAge = 35;
  cout << "I am " << myAge << "
years old.";
  return 0;
}

Output:-

I am 35 years old.
```