

# IS1201: Programming & Problem Solving

## 3. Data Types - 1



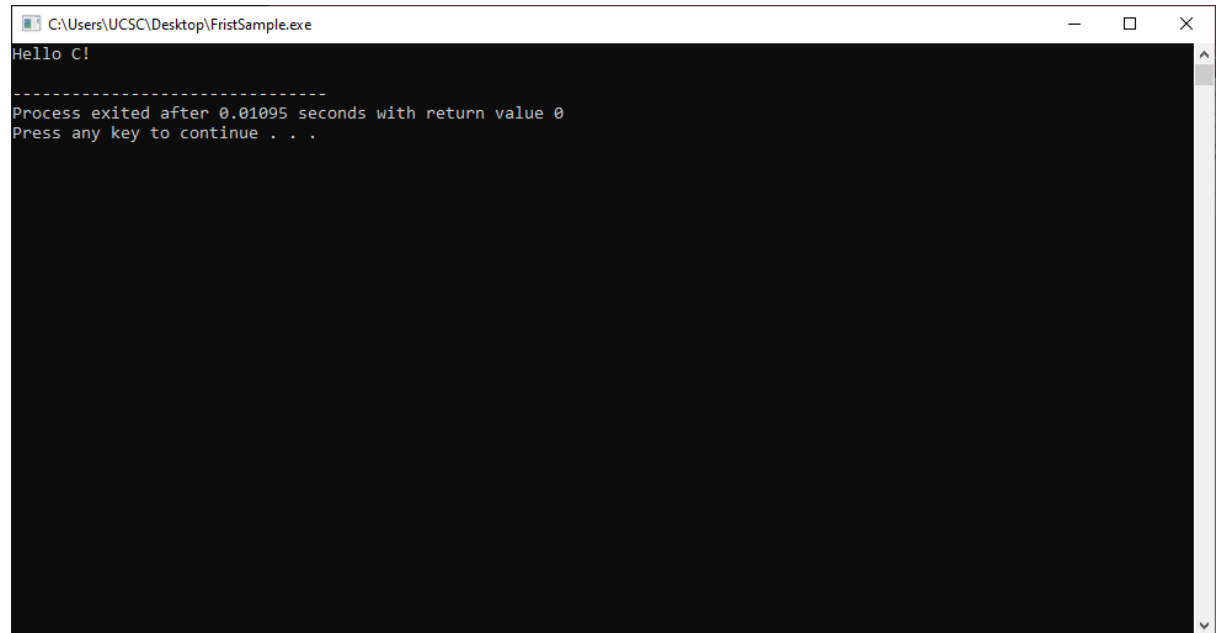
*Viraj Welgama*

# Your 1<sup>st</sup> C Program...

```
1  #include <stdio.h>
2
3  int main() {
4      printf("Hello C!\n");
5      return 0;
6  }
7
```

# Your 1<sup>st</sup> C Program...

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello C!\n");
5     return 0;
6 }
7
```



```
C:\Users\UCSC\Desktop\FristSample.exe
Hello C!

-----
Process exited after 0.01095 seconds with return value 0
Press any key to continue . . .
```

# Your 1<sup>st</sup> C Program...

```
1  #include <stdio.h>
2
3  int main() {
4      printf("Hello C!\n");
5      return 0;
6  }
7
```

- In this program, we have used printf() function which displays the text inside the quotation mark. Since printf() is defined in stdio.h, you need to include stdio.h.
- In C programming, the code execution begins from the start of main() function (doesn't matter if main() isn't located at the beginning).
- The code inside the curly braces { } is the body of main() function. The main() function is mandatory in every C program.

# Your 1<sup>st</sup> C Program...

```
1  #include <stdio.h>
2
3  int main() {
4      printf("Hello C!\n");
5      return 0;
6  }
7
```

- The printf() is a library function that sends formatted output to the screen (displays the string inside the quotation mark).
- Notice the semicolon at the end of the statement.
- The return statement return 0; inside the main() function ends the program. This statement isn't mandatory. However, it's considered good programming practice to use it.

# Key Notes...

- All C program starts from the **main()** function and it's mandatory.
- You can use the required header file that's necessary in the program.
- C is case-sensitive; the use of uppercase letter and lowercase letter have different meanings.
- The C program ends when the program encounters the return statement inside the main() function.
- The statement in a C program ends with a semicolon (;).

# Built in Libraries...

- C programming has several in-built library functions to perform input and output tasks.
- Two commonly used functions for I/O (Input/Output) are `printf()` and `scanf()`.
- The `scanf()` function reads formatted input from standard input (keyboard) whereas the `printf()` function sends formatted output to the standard output (screen).

# C Output

```
#include <stdio.h> // this is needed to run printf()

int main() {
    printf("C Programming"); //displays the content inside quotation
    return 0;
}
```

```
#include <stdio.h>
```

```
int main() {
    int testInt = 10;
    printf("Number = %d", testInt);
    return 0;
}
```

- Inside the quotation of printf() function, there is a format string "%d" (for integer). If the format string matches the argument (testInt in this case), it is displayed on the screen.



# How this program works?

- All valid C program must contain the `main()` function. The code execution begins from the start of `main()` function.
- The `printf()` is a library function to send formatted output to the screen.
- The `printf()` function is declared in "`stdio.h`" header file.
- Here, `stdio.h` is a header file (standard input output header file) and `#include` is a preprocessor directive to paste the code from the header file when necessary.
- When the compiler encounters `printf()` function and doesn't find `stdio.h` header file, compiler shows error.
- The `return 0;` statement is the "Exit status" of the program. In simple terms, program ends.

**When I wrote this code,  
only God & I understood what it did.**



**Now...  
only God knows.**

# Comments

- As programs get bigger and more complicated, they get more difficult to read. Formal languages are dense, and it is often difficult to look at a piece of code and figure out what it is doing, or why.
- For this reason, it is a good idea to add notes to your programs to explain in natural language what the program is doing.
- These notes are called comments , and they start with The `//` symbol or `/* */` for C.

# Problems

1. Write a C program to add two Integers
2. Write a C program to read and print your name

# C Integer Input/Output

```
#include <stdio.h>

int main() {
    int testInt;
    printf("Enter an integer: ");
    scanf("%d", &testInt);
    printf("Number = %d", testInt);
    return 0;
}
```

- The `scanf()` function reads formatted input from the keyboard. When user enters an integer, it is stored in variable `testInt`.
- Note the '`&`' sign before `testInt` gets the memory address of the `testInt` variable and the value is stored in that address.