CS-2060 Technical Documentation

GE02 – January 29, 2024

Table of Contents

Important Resources:

- C Standard Library Reference
- Programming Errors in C
- Format Specifiers in Java

Important Concepts to Remember:

Declaring Vs. Initializing Variables in C:

- **Declaring a variable:** Specifying its data type and name so that the compiler knows what kind of data the variable will store and how much memory to allocate for it.
  - `Int number; float pi;`
- **Initializing a variable:** Assigning an initial value to the variable at the time of declaration or a later point in the program.

    - `Int number = 42; float pi = 3.14;`
- It is **good** practice to initialize variables when declaring them in order to avoid your variable being filled with “garbage” data (data left over from some previous action).

Format Specifiers:

- %% → inserts a % sign
- %t %T → time and date
- %s %S → string
- %n → inserts a newline character
- %f → floating-point (decimal)
- %d → Decimal integer
- %c → character
- %b %B → boolean

Important Function Concepts:

Main:

- In C, the program’s “**main**” function indicates that the program has been executed successfully. This function is the entry point of the program, and its return value is typically used to communicate the exit status of the program to the operating system.
- `Return 0; //indicates a successful execution.`

Scanf:

- **Input Buffer Overflow**
    - Scanf in C reads input from the standard input/output stream <stdin>. If scanf fails to read or convert the given input into the data type defined by the format specifier, the value will be pushed into the buffer.
- **Infinite Loops**
    - If a program pushes values into the input buffer and fails to clear the input buffer, it could lead to a repeated loop making your code stuck.

Errors in C Programming:

- Javatpoint.com “Programming Errors in C”

Syntax error:

- Occurs at compilation time, thrown by the compiler.
- `Int a; //causes error, int a; //correct form`

Run-time error:

- Occurs after compilation and during the execution of a program.
- `Int a = 2; in b = 2/0; // Causes error b/c dividing by zero`

Linker error:

- Caused when the executable file of the program is not created.
- If the wrong function prototyping or usage of the wrong header file.

Logical error:

- An error that leads to an undesired output but is “error-free” (in terms of what the IDE recognizes.
- `For(int i=1; <=10;i++); {} //Causes error as we put the semicolon after the loop`

Semantic error:

- Occurs when a statement in the code is not understood by the compiler.
- `int i; i=i+2; // Causes error since i was never initialized`