TRACY WANGARI MUNGAI

SCT222-0341/2023

PROGRAMMING ASSIGNMENT

1. Define the following terms as used in C programming:

* Compiler – This is a software tool that translates the human-readable source code.
* Source code – This is the human-readable, high level programming instructions written by a programmer.
* Object code – Refers to the output generated by a compiler after translating the human-readable source code into a lower level representation.

1. Using an example i.e, a program to add 2 numbers, explain the compilation process of a C program.

* Editing: The programmer writes the source code in a text editor. For instance, a program to add two numbers might look like this:

#include <stdio.h>

int main() {

int num1 = 5;

int num2 = 7;

int sum = num1 + num2;

printf("Sum: %d\n", sum);

return 0;

}

* Preprocessing: The preprocessor (part of the compiler) handles directives starting with #. It might include files specified with #include and perform macro replacements.
* Compilation: The compiler translates the preprocessed code into assembly code or an intermediate representation.
* Assembly: The assembler converts the assembly code into machine code (object code) specific to the target architecture.
* Linking: The linker combines the object code with other necessary object code (like libraries) to produce an executable file. It resolves addresses and creates the final binary.
* Execution: The user runs the compiled program, and it adds the numbers during runtime, displaying the result as specified in the printf statement.

1. Explain the differences between a compiler and an interpreter.

* Translation Process:

Compiler: Translates the entire source code into machine code or an intermediate code before execution.

Interpreter: Translates the source code line by line and executes it immediately without creating a separate compiled version.

* Output:

Compiler: Generates an executable file or some form of intermediate code that is executed later.

Interpreter: Does not produce a separate executable; it directly executes the source code.

* Execution:

Compiler: The compiled code is executed independently of the original source code.

Interpreter: The source code is interpreted and executed directly during runtime.

* Performance:

Compiler: Generally produces faster execution as the entire code is translated before execution.

Interpreter: Might have a slower execution since it translates and executes code line by line.

* Error Handling:

Compiler: All errors need to be fixed before the compilation process starts.

Interpreter: Can catch and report errors as it encounters them during the execution of the code.

* Memory Usage:

Compiler: Generally requires more memory as it produces a separate compiled version of the code.

Interpreter: Tends to use less memory since it executes code directly without creating a compiled version.

* Portability:

Compiler: The compiled code is often specific to the target machine architecture.

Interpreter: The same source code can be run on any machine with the appropriate interpreter installed.

1. List all main categories of operators available in C programming.

* Arithmetic Operators:

+ (Addition)

- (Subtraction)

\* (Multiplication)

/ (Division)

% (Modulus or remainder)

* Relational Operators:

== (Equal to)

!= (Not equal to)

< (Less than)

> (Greater than)

<= (Less than or equal to)

>= (Greater than or equal to)

* Logical Operators:

&& (Logical AND)

|| (Logical OR)

! (Logical NOT)

* Assignment Operators:

= (Simple assignment)

+= (Addition assignment)

-= (Subtraction assignment)

\*= (Multiplication assignment)

/= (Division assignment)

%= (Modulus assignment)

&= (Bitwise AND assignment)

|= (Bitwise OR assignment)

^= (Bitwise XOR assignment)

<<= (Left shift assignment)

>>= (Right shift assignment)

* Increment and Decrement Operators:

++ (Increment)

-- (Decrement)

* Bitwise Operators:

& (Bitwise AND)

| (Bitwise OR)

^ (Bitwise XOR)

~ (Bitwise NOT)

<< (Left shift)

>> (Right shift)

* Conditional (Ternary) Operator:

? : (Conditional operator)

* Miscellaneous Operators:

Size of (Size of a variable or data type)

, (Comma operator)