INTRODUCTION TO C PROGRAMMING

J Kamiri

What is C programming Language?

- C is a powerful computer programming language that is appropriate for technically oriented people with little or no programming experience
- C evolved from two previous languages, BCPL and B.
- BCPL was developed in 1967 by Martin Richards as a language for writing operating-systems software and compilers.
- B was developed by Dennis Ritchie at Bell Laboratories
- Different versions of C programming were develop as the language evolved. This brought challenges and necessitated the creation of C standard library

What is the C standard Library

- It is a collection of built-in functions that a C programmer can use to perform certain functions.
- These functions make it easy for the programmer to develop a programmer with few lines of code.
- Also, the functions contained in this library are numerous, therefore, a developer only interacts with the functions they need to solve the task at hand.
- Commonly used functions include
  - Printf()- for outputting
  - Scanf()- for inputing
- Every C program must start by including the standard library using this line of code *(#include<stdio.h>)*

Cont.

- Every programming language is composed of elements that contribute to the language’s syntax and semantics.
- These elements include:
  - Datatypes
  - Identifiers
  - Functions
  - Arrays
  - Operators Etc
- A thorough understanding of these elements helps a programmer to easily understand the logic of a program.

Data Types in C Programming

Data Types in C programming

Datatype is a classification that defines the type of data that a variable holds. It tells the compiler or interpreter how the computer intends to use the data.

Data types can also be broadly classified into:

- **Numeric data types:** Contain numeric values i.e numbers
- **Non-numeric datatypes:** contain non-numeric values i.e alphabets

The data types in C can be further classified as follows

- **Basic Types:** They are arithmetic types and consists of the two types: (a) integer (b) float (c) Double (d) character

Basic datatypes

Basic datatypes are the most commonly used datatypes in C programming

- **char:** The most basic data type in C. It stores a single character and requires a single byte of memory in almost all compilers. E.g ‘A’ , ‘B’
- **int:** As the name suggests, an int variable is used to store an integer. i.e whole numbers e.g 1, 2, 10
- **float:** It is used to store decimal numbers (numbers with floating point value) with single precision (occupies 32 bits of memory). E.g 1.1
- **double:** It is used to store decimal numbers (numbers with floating point value) with double precision (occupies 64 bits of memory). E.g 4.5
- **Boolean:** It is a binary datatype i.e true/ false or 1/0

Data Types

- **Enumerated types:** They are again arithmetic types and they are used to define variables that can only be assigned certain discrete integer values throughout the program. E.g enum datatype
- **The type void: The** type specifier *void* indicates that no value is available. E.g (void)
- **Derived types:** They include (a) Pointer types, (b) Array types, (c) Structure types, (d) Union types and (e) Function types.

Each datatype has its own range

See the attached list of datatypes in C programming and their ranges. Note that basic datatypes are the most commonly used datatypes.

Expressions

- An expression is a sequence of operands and operators. Operands are values reacted upon.
- Operators are symbols that tell the computer to perform certain mathematical or logical operations.
- For example (x+y=z).
  - X, y, and Z are operands while + and = are an operator.
- Operators can be grouped into the following categories/ types
  - Arithmetic
  - Logical
  - Assignment
  - Comparison

Arithmetic operators

- Arithmetic operators-they carry out mathematical operations
- C follows BODMAS in executing an equation

Symbol	**Meaning**
*	Multiplication
/	Division
%	Modulus or Remainder
+	Addition
^	Exponential (e.g x^2)
-	Subtraction

Comparison/ Relational Operators

- Used to show the relationship between operands

Operator	**Description**
==	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
!=	Not equal to

Assignment Operators

Operator	**Example**	**Equivalent**
+=	bonus+=500	bonus=bonus+500
-=	Budget	budget=budget-50
*=	salary*=1.2	salary=salary*1.2
/=	factor/=.50	factor=factor/.50
%=	daynum%=7	daynum%7
=	X=0	Means the value of x is 0

Logical operators

- They are used to test more than one condition and make a decision. They always perform true or false results.

Operator	**Meaning**
&&	AND
||	OR
!	NOT

Special Operators

Operator	Example	Description	Equivalent Statements
++	i++	Postfix incrementing	i=i+1;i+=1;
++	++I	Prefix incrementing	i=i+1;i+=1
– –	i- -	Postfix decrementing	i=i-1;i-=1;
– –	- -i	Prefix decrementing	i=i-1;i-=1;

Program Writing

IDE Setup

- IDE stands for Integrated development Environment
- It is as software the is used to write and execute programs
- A good IDE should have the following:
  - ✓An Editor- to write your code
  - ✓A debugger- to debug your codes. Real time debuggers detect syntax errors when you are writing your code.
  - ✓A complier- Translates source code to machine code
  - ✓Libraries- collection of files, programs, routines, scripts, or functions that can be referenced in the programming code
- Some of the suitable IDE for C programming include: Netbeans, DevC++, Codeblocks, Visual Studio

PROGRAMMING ERRORS/BUGS

- **Syntax error-**it results as a violation of rules and procedures of writing a program e.g omission of a semicolon
- **Semantic errors-**It is the violation of the meaning inscribed in a particular term.
- **Logical error-**program is running properly but giving undesired results or doesn’t make sense.
- **Compilation/Run-time/Execution time error-**This error happens when compiling a program e.g division by zero i.e 5/0

Phases of a C program

C program goes through six phases

Editing-This is accomplished with an editor program e.g Ms Visual studio, Borland, C++ builder e.t.c

- The Programmer types a c-program with the editor, makes correction if necessary, then stores programs on a secondary storage device such as a disk. C-program file names should end with the .c extension

Pre-Process-The processor program executes automatically before the compiler translation phase begins.

- The C-preprocessor directives which indicate that certain manipulations are to be performed on the program before compilation.

Compilation phase-

- A computer only understands the machine language, therefore, every program written in any other language needs to be translated to the machine language.
- Different programming languages use different tools. These tools include:
  - Translator
  - Compiler
  - Assembler
- This is the phase where source code is converted to machine code.
- The programmer gives the command to compile the program. The compiler translates the high level language (source code) into machine language (object code).

Cont.

- An *interpreter* translates a source program word by word or line by line. This allows the CPU to execute one line at a time.
- The Interpreter takes one line of the source program, translates it into a machine instruction, and then it is immediately executed by the CPU.
- A ***compiler*** translates the entire/whole source program into object code at once, and then executes it in machine language code. These machine code instructions can then be run on the computer to perform the particular task as specified in the high-level program.
- An ***assembler*** translates programs written in Assembly language into machine language that the computer can understand and execute.

Cont.

- **Linking-**C-programs typically contain differences to functions defined elsewhere such as in the standard library or in private libraries groups of programmers working on a particular project.
- Linker-Links the object code with the source code to ensure no missing functions to produce an executable program(no missing pieces)
- **Loading phase-**Before the program can be executed the program must first be placed in the memory, this is done by the loader.
- Loader takes the executable program from the disk and transfers it to the memory. Additional components from shared libraries that support the program are also loaded.

Executing phase-The compiler under the control of its control processing unit, executes the program/instructions at a time.

NB: Programs don't always work on first try. Each of preceding phases can fail due to various errors.

STRUCTURE OF A C PROGRAM

```
#include<stdio.h>//pre-processor directive
 Int main ()// main function
{
Printf("My first program\n");
Return 0;/*program end successfully*/
}
```

Parts of a Program

- Documentation section-it consists of comments giving the title of the program, the author, the date of creation, the aim or objectives of the program.
- The compile directive-It provides instructions to the compiler to link the functions to its system library e.g #include<stdio.h>,#include<math.h>.#include<stdlib.h>
- Main function-It indicates the start of execution or the program statement. The main function is used to call any other program in the body of a program and is executable in a sequential manner. A function is a small program that carries out a specific task when called

Parts of a Program Cont.

- Body of the program-It consists of opening and closing braces.
- Declaration of variables-All variables must be declared before use.
- Return 0-This means terminating a function, it puts an end to execution of a program. It returns the control to the operating system

GOOD PROGRAMMING PRACTICES

Braces({ })-The opening and closing braces. All braces must be opened and must be closed

Whitespace-it refers to character in programming tabs, newlines etc.

Indenting-It refers to moving away from the margin

Good programming practices

Use of comments-comments are non-executable statements in a program that tries to tell what part of the program is doing or give the name(author of the program, date of creation or explain a block,a single statement, a function or intended purpose)

Advantages of using comments

- It makes a program understandable (readability)
- It makes the modification of a program more easier in future

Two types of comments

- Single line comment (//)-its used when commenting using one line

  e.g //My first program in C

  Multiple line comment/*---------------------------------

  */

- Its used when commenting on multiple or many lines also known as block comment

  e.g/*Program to calculate the total, and average of 5 units*/

Use Escape sequence-These are programming shortcuts that are used when writing codes and they perform a certain task. Its represented by a backslash followed by an escape character

Escape Sequence

Escape Sequence	**Description**
\’ (single quote)	Outputs the single quote character
\”(double quote)	Outputs the double quote character
\? (Question mark)	Outputs the question mark character
\\(backslash)	Outputs the backslash character
\a (alert or bell)	Cause an audible bell or visual alert
\b (backspace)	It moves the cursor back on position on the current line

Cont.

\f (new page or form feed)	It moves the cursor to the start of the next logical page
\n (new line)	It moves the cursor to the beginning of the next line
\r (carriage return)	It moves the cursor to the beginning of the current line
\t (horizontal tab)	It moves the cursor to the next horizontal tab position
\v (vertical tab)	It moves the cursor to the next vertical tab position

Writing a Simple C program to output Welcome to MUT

```
#include <stdio.h>
int main() {
	printf("Welcome to MUT");


	return 0;
}
```

Example 2

```
#include <stdio.h>
int main() {
printf("Welcome to MUT\n I am happy about it");


        return 0;
}
```

- The program here shows how to output in two lines using a single output statement.
- The output here is expected to be

```
Welcome to MUT
I am happy about it
```

Example 2. Cont.

- The same program in Example 2 (previous page) can be written as shown here.
- Note that in this case we have used two output statements rather than a single output statement.
- This is a redundant way of outputting which complicates a program

```
#include <stdio.h>
int main() {
printf("Welcome to MUT\n");
printf("I am happy about it");
        return 0;
}
```

Writing a Simple C program to output Hello World

```
#include <stdio.h>
int main() {
        printf(" Hello World \n");
        return 0;
}
```

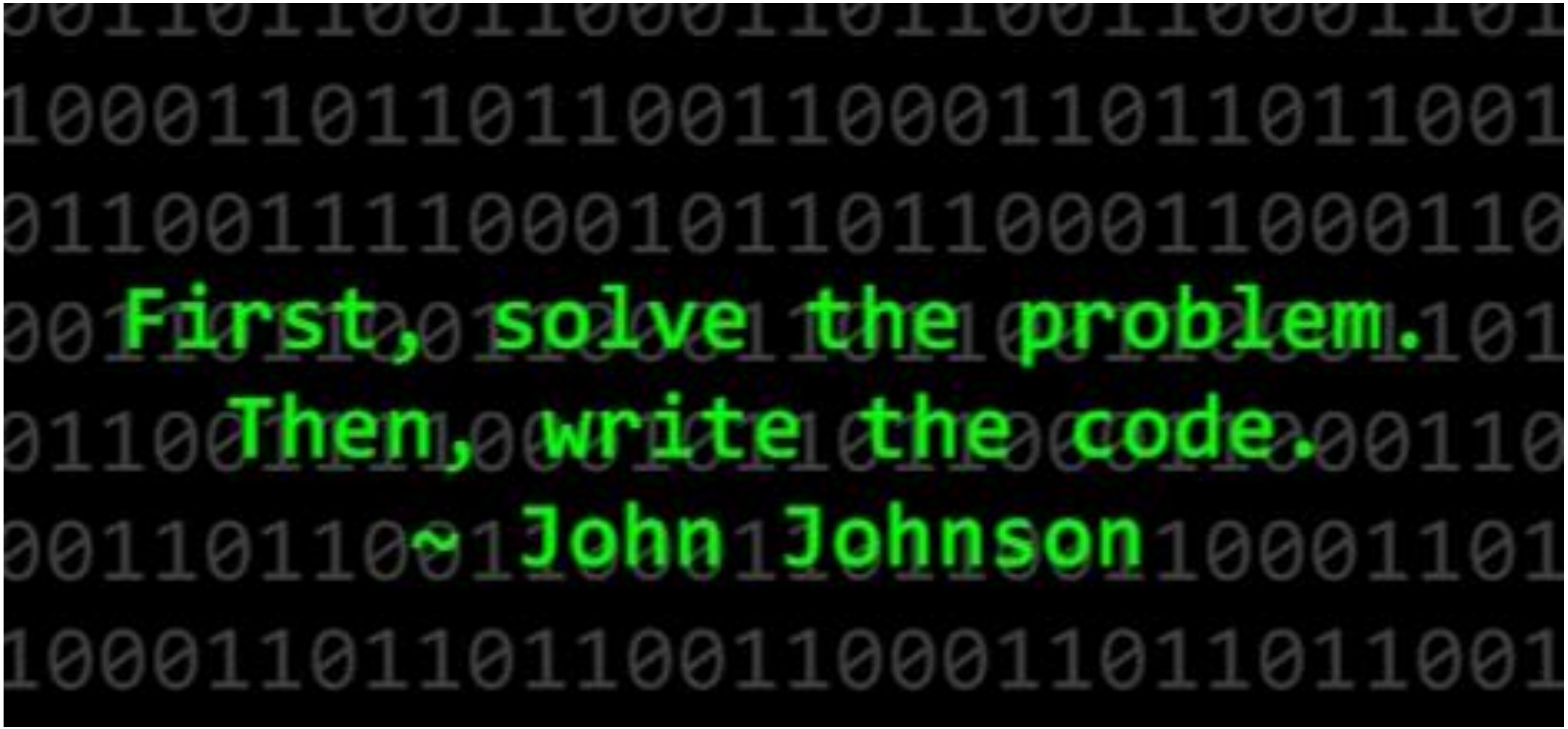
Task

- Write a C program to output your name and your home county as shown below.

  My Name is ……..

  I am from ……. County

The End



First, solve the problem. Then, write the code. ~ John Johnson