**NAME: ADEYE INIOLUWA BOLUWATITO**

**MATRIC NUMBER: 222454**

**COURSE: CSC 235**

1. A brief note on Unix OS especially UNIX flavor.

The Unix operating system is a set of programs that act as a link between the computer and the user. Unix is called a multiuser system which means it can be used at the same time by many people; it is also called a multitasking environment because multiple programs can be run at the same time by a user. Examples are AUX, HP UNIX and BSD. Linux is also a flavor of Unix which is readily and freely available.

Let’s first of all explain the term “flavors of Unix”. This refers to many UNIX-like operating systems developed based on the original UNIX written by Ken Thompson in 1969 at Bell Labs. Linux, being one of the flavors of UNIX was developed by Linus Torvalds which is an open source UNIX-like operating system that mimics the form and function of a UNIX system which was developed independently and not derived from licensed source code.

1. Software functional requirements.

Software requirement is a condition or capability possessed by a software or system component in order to solve a real problem or achieve an objective by a user. Commonly considered requirements are classified into 3 categories namely, functional, non-functional and domain requirements. We will be focusing solely on functional requirements.

Functional requirements describe the interaction of software with its environment and specify the inputs, outputs, external interfaces, and the functions that should be included in the software. Functional requirements should be complete and consistent. It should be complete which implies that all the user requirements are defined; and consistent which means that all requirements are specified clearly without any contradictory definition. Functional requirements define what a software component must do. Preparing functional requirements can be done in a number of ways but the most common way is that they are documented in a text form. Examples of functional requirements are authentication, certification requirements, audit tracking, etc.

1. Unix is often prefered at some point because it is a multi-user tasking operating system and it also uses less memory while running sophisticated programs. It can also handle virtual memory nicely. Many developers also prefer Unix due to the developer-friendly environment, core security features, portability and performance.
2. Unix is being referred to as a scientist operating system because it is well equipped with developer friendly tools like debugging tools, type checking tools, etc which automate away many of the little tasks that would otherwise distract the developer form concentrating on the most important aspect of development which is his / her design.
3. C is a procedural programming language which means it follows a sequence of statements in order to achieve a desired output. Based on conversion to make it readable by a computer, C is a compiled programming language whereby the entire program is converted into machine code all at once.
4. Detailed structure of a complete C programming language.

The structure comprises of Documentation, Link, Definition, Global Declaration, Main() function and subprograms.

Example of a detailed C programming language:

    /\* A program to calculate the Body Mass Index (BMI) and the corresponding

    category –

under-weight, normal and over-weight. \*/ ***// Documentation***

#include <stdio.h>

#include <stdlib.h> ***//Link***

#include <string.h>

#include <math.h>

#define STATUS Uncategorized ***//Definition***

double body\_mass\_index(double weight, double height); ***//Global declaration***

int main(){ ***//Main() function***

    printf("A program to calculate Body Mass Index (BMI) and the state the corresponding category either under-weight, normal and over-weight.\n");

    double user\_weight;

    double user\_height;

    printf("Enter your weight in kilograms(kg): ");

    scanf("%lf", &user\_weight);

    printf("Enter your height in metres(m): ");

    scanf("%lf", &user\_height);

    // calculation of the minimum number of kg to lose or gain

    double lose\_kg = body\_mass\_index(user\_weight, user\_height) - 18.5;

    double gain\_kg = body\_mass\_index(user\_weight, user\_height) + 18.5;

    // RESULT

    printf("\nYOUR RESULT:\n");

    printf("Your body mass index is %.2f and ", body\_mass\_index(user\_weight, user\_height));

    if(body\_mass\_index(user\_weight, user\_height) >= 25.0){

        printf("You are overweight. It is advisable to lose a minimum number of %.2f kg to be in the normal category.\n", lose\_kg);

    } else if(body\_mass\_index(user\_weight, user\_height) < 18.5){

        printf("You are underweight. It is advisable to gain a minimum number of %.2f kg to be in the normal category.\n", gain\_kg);

    } else{

        printf("Congratulations! You have a normal weight\n.");

    }

    return 0;

}

/\* the function for body mass index \*/

double body\_mass\_index(double weight, double height){ ***//Subprograms***

    double user\_body\_mass\_index = weight / (pow(height, 2));

return user\_body\_mass\_index;

}

1. How to create a C programming file on the OS.

* Open the linux terminal.
* Using a text editor e.g vscode, create file using editor file\_name.c.