**NAME:** AYINLA FAOUR TEMILOLUWA

**LEVEL:** 200

**DEPARTMENT:** COMPUTER SCIENCE

**MATRIC NUMBER:** 222466

**COURSE:** CSC235

***ASSIGNMENT***

1. Unix which stands for Uniplexed Information Computing System is the first operating system that was implemented using C language and developed by Ken Thompson and Dennis Ritchie at AT&T Bell laboratories in 1970. It is a multi-user multi-tasking system for computers.

There are many versions of UNIX and they share some similarities, here are examples of UNIX versions: Sun Solaris, MacOS X, GNU/Linux. The Unix OS is broken into 3 parts – Kernel, Shell and the programs. There are Linux flavors, here are some of them:

* Ubuntu – was created by canonical in 2004. It was created to be used as a graphical linux desktop without using command line. It is the most popular linux distribution.
* Fedora Linux - this linux distro is focused on free software and latest versions of software. It’s less stable than other linux flavors, but it provides the most up to date software.
* Linux Mint - this was built with Ubuntu as its foundation. It’s similar to Ubuntu in that some software packages are common in both. It uses cinnamon and mate desktop instead of Ubuntu desktop environment.
* openSUSE – is similar to Fedora but its more stable and older.
* Debian – this one has been in existence since 1993 and is more stable than Ubuntu because it releases its versions slower than its counterparts. Ubuntu was made with Debian as its foundation so you can say Ubuntu is an improved version of Debian.
* Arch Linux – this is not a beginner friendly version because every package has to be installed by the user.

There are many other versions like Mandriva Linux, BlackArch, Peppermint OS and so on.

1. Software Functional Requirement is the description of the service a software is being built for. It’s a set of standards or requirement that a software is created to meet or better still what solution a software provides.

The functional Requirement of a Software must include the following:

* Complete Information about the workflow of the software.
* Details of every operation conducted by the software.
* It should clearly define who will be allowed to modify/create/delete data in the system.
* It should have one or more descriptions of system reports or Outputs.

Software Functional Requirement has the following benefits:

* Gives you an overview as to whether the software is providing all functionalities shown in the requirements.
* Any Error noticed at the stage of functional requirement is usually one of the cheapest to fix.
* It also helps to identify missing requirement in the software systems.

There are various types of functional requirements:

* Transaction Handling
* Audit Tracking
* Administrative Functions
* Certification Requirements
* Historical Data Management
* Business Rules

Examples of Functional Requirements are:

* A sales system should allow users to record customer sales
* Only managerial level employees have the right to view revenue data
* Software system should be integrated with banking API

1. Unix is preferred to at some point because it is developer friendly as it is open-source. The open source movement of the .NET framework is what makes it preferred to.
2. The special characteristic of Unix being open source is what has attracted a lot of scientists to it and it makes it easier for scientists to manipulate software and computers to achieve a goal.
3. C programming language is a high-Level compiled programming language as it uses a compiler to compile the code written.
4. Here’s the structure of a C program file:

* Document Section - this is where you comment on your program using /\* \*/
* Preprocessor/ link section – this includes the header files #include <stdio.h>
* Definition Section – this is the section you define variables in your code.
* Global Declaration Section – declare or define global variables.
* Function declaration section – gives information about a function that includes datatype or return type.
* Main function – the section where your main code goes into.
* User-defined Function - to define a function to fulfill a particular requirement.

1. You can create a C program file on UNIX OS by following these steps:

* Install compiler and other Dev tools. we can install on Ubuntu or Debian systems. Type commands **sudo apt-get update** and **sudo apt-get install build-essential manpage-dev** into your terminal
* Check GCC version **gcc –v**
* Open text editor on Ubuntu. We can use nano or vim or gedit. Let’s use nano. To install nano we type in command **sudo apt install nano**
* Write your first C program in the linux terminal. Create a file:
* nano demo.c
* write your code
* Compile with GCC. Type this into the GCC compiler **gcc demo.c -o demo**

We are compiling the demo.c file using GCC and saving it as demo

* Run C program in Ubuntu terminal

./output-program-name

In this case would be ./demo