NAME : AJAKAYE JADESOLA STELLA

MATRIC NUMBER : 18/SCI01/010

COURSE CODE: CSC206 Revision

1) i) **Programming**: It is the process of writing programs.

 ii) **Programs**: A program is a set of codes that instructs the computer to carry out some processes.

iii) **Programming languages**: They are the languages through which we instruct the computer to carry out some processes or tasks.

2) **Types of Programming Languages:**

1. Machine Language: Machine language or machine code is the native language directly understood by the computer’s central processing unit or CPU. It is not easy to understand , its uses binary system.
2. Assembly Language: It is language which consists of symbolic codes and which are easier to remember than machine codes.
3. High-Level Language: They are user friendly languages which are similar to English symbols with vocabulary of words and symbols. They are easier to learn and requires less time to write.

 3) **Features of Programming Languages:**

1. Language structure which consists of keywords, expressions and statements.
2. Translator is required before being understood by the computer.
3. Programs are written for the purpose of communicating data between human and the computer
4. Syntactic rules for forming statements.
5. Vocabulary that consists of letters of the alphabets.

4)  **Programming Methodologies:**

a)  **Procedural Programming**: It is a series of steps, each of which performs a calculation, retrieves input, or produces an output. e.g. Assignment, loops, sequences

b) **Object Oriented Programming**: It is the collection of objects that interact with each other by passing messages that transforms their state. E.g. Python, Java.

c) **Functional Programming**: It is a collection of mathematical functions, each with an input (domain) and a result (range).E.g. LISP, HASKELL.

d) **Logic Declarative Programming:** It is a collection of logical declaration about what action a function should accomplish rather than how that outcome should be accomplished. E.g. PROLOG.

e) **Event Driven Programming**: It is a continuous loop that responds to events that are generated in an unpredicted order. E.g. VB & JAVA.

f) **Concurrent Programming**: It is a collection of cooperating processes, sharing information with each other from time to time but generally operating asynchronously .E.g. LINDA & High FORTRAN.

5) **Five major steps in developing an efficient program:**

a) **Problem Analysis:** This is where a clear statement of the problem is stated

b) **Design:** The planning of the solution to the problem in the first stage takes place in this stage

c) **Coding:** Translation of the algorithm in stage two into a programming language takes place here.

d) **Testing and Debugging:** This process involves the location and removal of errors in the program if any.

e) **Documentation:** This is the final stage of the program development, it consist of organizing all the materials that describes the program.

6) **Consideration of writing a good programs:**

a**) Naming Conventions:** It is very important to give meaningful name to all your constructs.

b) **File naming and Organization**: Files should be organized into directories instead of having monolithic structure where all source code files are and all header files are in a single directory.

c) **Formatting and Indentation**: The lines within the code should be clearly organized in such a way it can easily be read and understood by the writer.

d) **Comments and Documentation**: Introducing comments and proper explanations (documentation) of the program aids in understanding the code.

e) **Classes**: Ensure that all classes in your application must have a default constructor, copy constructor and overload operator .

f) **Functions:** They are meant to do only one job and execute it well.

g) **Using STL**: Using Standard Template Library instead of creating your own container data structures.

h) **Minimizing bugs by testing**: Testing is an integral part of software development. It helps in preventing errors.

7) **What is Structured Programming**? Structured Programming is a programming paradigm aimed at improving the clarity, quality and development time of computer program by making extensive use of subroutines, block structures, for and while loops.

8) **Fundamental Objectives of Structured Programming**: a) To increase programmers productivity.

b) To increase program clarity by reducing complexity.

c) To reduce program testing time

d) To reduce program maintenance time and effort

9) **Logical pattern that characterize structure programming**:

a) **Sequence**: It refers to an order of execution of statements.

b) **Selection**: It uses conditions and one of a number statements is executed depending on the state of the program.

c) **Repetition**: A statement is executed until it reaches a certain state, or operations have been applied to every elements of a collection.

10) **Method Employed**: ALGORITHM

 Properties required for algorithm:

1. Input
2. Output
3. Correctness
4. Definiteness
5. Finiteness
6. Generality

11) An algorithm using flowchart to find the volume of a sphere, given radius as 23

 Volume of a Sphere = $\frac{4}{3}πr3$

 Radius = 23

 **Algorithm:**

1. Start the execution.
2. Read the value of R which equals 23.
3. Calculate the Volume. [i.e 4/3\*$π$\*r\*r\*r ]
4. Print the value of V.
5. End the execution.

 **Flowchart:**

 Read R

V = $\frac{4}{3}πr3$

 Print V

12) An algorithm using Pseudocode to find the radius of a cone, if the volume is given as 200cm3 and height as 18 .

 **Algorithm:**

Pseudo code;

1. Start
2. Input Volume
3. Input Height
4. Radius = $\sqrt{\frac{3V}{πH}}$
5. Print “ The radius is “,Radius.
6. End

13) Differences between Object Oriented Languages and Structured oriented Languages.

|  |  |
| --- | --- |
| OBJECT ORIENTED LANGUAGES | STRUCTURED ORIENTED LANGUAGES |
| It is designed which focuses on data | It is designed which focuses on process |
| It follows bottom up approach | It follows top-down approach |
| It is divided into small entities called OBJECTS | It is divided into small self contained FUNCTIONS |
| More abstraction and more flexibility | Less abstraction and less flexibility |
| Less function dependency | More function dependency |

14) Describing each line in the program below

#include<stdio.h> (Link section)

#define PI 3.142 (Definition section)

int main() (Main function)

{

 const float rad =7.5;

 float cir;

 cir = 2\* PI \* rad; (User function defined)

 printf (" the value of circumference of a circle is = %d",cir);

 return 0;

}