Name: OHOBORE OGHENEKEVWE EMMANUEL

Matric Number: 18/sci01/064

CSC206 PRACTICE QUESTION I

1.i.Programming : programming is the process of designing and building an executable computer program to accomplish a specific computing result.

ii.Program: A program is a collection of instructions that can be executed by a computer to perform a specific task.

iii.Programming languages: A programming language is a formal language comprising a set of instructions that produce various kinds of output.

2.(a)Machine Language: Machine language is the lowest-level programming language. Machine language is a collection of binary digits or bits that the computer reads and interprets. Machine languages are the only languages understood by computers.

(b)Assembly Language: An assembly language is a low-level programming language designed for a specific type of processor.

(C)High-level Languages: High level language is anything machine independent.

3.Features of programming

 A.It must have syntactic rules for forming statements.

 B.It must have a vocabulary that consists of letters of thge alphabet.

 C.It must have a language structure, which consists of, keywords, expressions, and statements

 D.It may require a translator before it can be understood by a computer.

 E.Programming languages are written and processed by the computer for the purpose of communicating data between human being and the computer.

4.Programming Methodologies

 I.Procedural programming: It is a series of steps, each of which performs a calculation, retreives input, or produces output. Concepts like assignments, loops, sequences and conditional statements are the building blocks of procedural programming. Examples of procedural programming are COBOL, FORTRAN, C, and C++.

 II.Object-Oriented Programming: It is a collection of objects that interact with each other by passing messages that transform their state. The fundamental building blocks of Object-oriented programming are object modelling, classification, and inheritance. Major object=oriented languages are python, Java e.t.c.

 III.Functional Programming: This is a collection of mathematical functions, each with an input(domain) and a result(range). Interaction and compositions, conditionals and recursion. Major functional programming languages are Lisp, Scheme, Haskell, and ML.

 IV.Logic Programming: Logic program is a colecction of logical declarations about what outcome a function should accomplish rather than how that outcome should be accomplished. Logic programming provides a natural vehicle for expressing non-determinism, since the solutions to many problems are often not unique but manifold, The major logic programming language is prolog.

 V.Event Driven programming: An event driven program is a contionous loop that responds to events that are generated in an unpredictbale order. These evnts originate from user actions on the screen(mouse clicks or keystrokes, for example), or else form other sources(like readings form sensors on a robot). Major event-driven programming languages include Visual basic and java.

 VI.Concurrent Programming: A concurrent program s a collection of cooperating program is a collection of cooperating processes,sharing informaton with each othertime to time but grnerally operating asynchronously. Concurrent programming languages include Linda and High performance FORTRAN.

5. These are the five major stages we are going to take in developing an efficient program for johnson LTD;

 (A)Problem Analysis: We must understand the problem and how to solve it, what is expected of the problem, what the program should do, the nature of the output and the input to consider so as to get the output.

 (B).Design: In this stage we are going to find the algorithm in solving the problem. We are going to use these methods; flowcharts, pseudocode, and a top-down chart.

 (C).Coding: In this stage we are going to write the code in use to solving the problem of the program. We are going to use the algorithm devised in the design stage along with of programming language we choose.

 (D)Testing and debugging: This process involves us locating and remving he errors in the prgram if there is any.

 (E).Documentation: This is the final stage. We are going to describe or tell how the program works in a document.

6.When writing good programs, the following should be taken to into consideration;

 I.Naming coventions: It is very important to give meaningful names to all your constructs E.g A name like get\_Height() or get\_avg\_height() gives more information than ctunde().

 II.Comments and documentation: Introducing comments and proper explanations(documentation) of the program aid in understanding the code. They help us in following the program flow, and skip parts for which we are not interested in details. This allows for program amendment and extensiblity.

 III.Classes: Ensure that all th classes in oyur application have a default constructor, copy constructor and overload operator. Also ensure that all the class dtaiems are appropiately initialized in condtructor and assigned to each member of the class.

 IV.Functions:A function should normally do only one job and do it well. Avoid generic functions with lots of conditional branches to do everything. if a function is supposed to do multiple jobs, then create helper functions and delegate responsibilities to them. Make functions simple and small. The ideal size of functions is around 35-40 lines.

 V.Using STL: Use Standard Template Library (STL) instead of creting your own container data structures.

 VI.Minimizing Bugs by Testing: Testing is an integral part of software development. Tests help us not only in making sure that what we have written is correct, but also in finding out if someone breaks the code later. So it is a good programming style to thoughly test a program.

 7.Structured programming: It is a programming paradigm aimed at improving the clarity, quality and development tme of computer program by making extensive use of subroutines,block structures, for and while loops.

 8.Objectives of Structured programming;

 1.To increase programmers' productivity

 2.To increase program clarity by reducing complexity

 3.To reduce program testing time.

 4.To reduce program maintenance time and effort.

 5.Structured programming gives room for well-thought-out program logic and provides an attempt to keep programs as simple and straight forward as possible.

9.Logical pattern that characterize structured programming

 I.Sequence: Sequence refers to an ordered execution of statements. It has one entry and exit points.

 II.Selection: It uses conditions and one of a number of statements is executed depending on the state of the program. This is usually expressed with keywords such as then e.t.c

 III.Repetition: In repetition, a statement is executed until program reaches a certain state, or operations have been applied to every element of a collection. This usually keywords like While e.t.c

 10. The method i'm going to employ in solving the task to solve the time table schedule problem is Algorithm

 Properties of the algorithm

 1.Input 2.Output 3.Definiteness

 4.Correctness 5.Finiteness 6.Correctness

 7.Effectiveness 8.Generality

11.



12. Input Volume = 0.000002m^3

 Input Height= 18m

 Radius= ((3\*0.000002)/pi\*18)^1/2

 Print "Radius is 0.000362m", Radius

13. Structured Programming

1. Structured Programming is designed which focuses on process.
2. Structured programming follows top-down approach.
3. In Structured Programming, Programs are divided into small self contained functions
4. Structured Programming provides less reusability, more function dependency.
5. Less abstraction and less flexibility.

Object Oriented Programming

1. Object Oriented Programming is designed which focuses on data.
2. Object oriented programming follows bottom-up approach.
3. In Object Oriented Programming, Programs are divided into small entities called objects
4. Object Oriented Programming provides more reusability, less function dependency.
5. More abstraction and more flexibility.

14. // program to find the circumference of a circle. //Comment Section

#include //Link Section

#define pi 3.142 //Definiotion section

int main () //Main Function

{ const float rad = 7.5; float cir; cir = 2 \* pi \* rad;

printf(“the value of circumference of a circle is %d”, cir);

return 0; } //User defined section