csc 206 revision questions by Harim Bello Shehu 19/sci01/100

1(i). Programming can be defined as the implenmentation of logic to facilitate specified computing operations and functionality.

 or

It can also be defined as the process of writing computer programs.

1(ii). A program can be defined as the collection of instructions that can be executed by a computer to peform a specific task.

2. Types of programing languagues :

i. Machine language.

ii. Assembly language.

iii. High level Language.

 Machine language: This is the native language of a processor and the collection of binary digits that the computer reads and interprets.

 Assembly Language: is a human readable representation of human native language

 or

 A programming language that consists of instructions of instructions that are mnemonic codes for corresponding machine language instructions.

 High level programming: is a programming language such as C, FORTRAN, or pascal that enables a programmer to write programs that are more or less independent of a particular type of computer.

3. FEATURES:

Syntax and structure for commands can overlap just like when using words in spoken languages.

Functionality of languages can make the same functionality similar to how all spoken languages.

4. Procedural programming: problem is brokem down into procedures, or blocks of code that peform one task each. All procedures taken together form the whole program. It is suitable only for small programs that have low level of complexity.

Object-oriented programming: the solution revovles around entites or objects that are part of problem. The solution deals with how the enitites behave and how they interact with each other to give a cohesive solution.

Functional programming: The problem or the desired solution is broken down into the functional untits. each unit peform its own task and is self- sufficient. These units are then stitched together to form complete solution.

Logical Programming : Here the problem is broken down into logical units rather than functional units. for example; In a school management system, users have very defined roles like class teacher, subject teacher etc. So the sofware can be divided into units depending on user roles.

Top-down or modular approach; the problem is broken down into smaller units which may be further broken down into even smaller units. Each units is called a module. Each module is a self-sufficient unit that has everything necessary to pefrom its task.

Bottom-up Approach; in bottom-up approach system design starts with the lowest level of components which are then interconnected to get higher level components. This process continues till a hierarchy of all system components is generated.

6. In writing a good program;

i. You need to understand the problem you are trying to solve.

ii. Design a solution

iii. Draw a flow chart

iv Write a pseudocode

v . Write code

vi. Test and debug

vii. Test with real world users

viii. Dont repeat code

viii. Avoid writing long code lines.

7. Structured programming is a programming paradigm aimed at improving the clarity quality and development time of a computer program by making extensive use of the structured control flow constructs of selcetion and repetition, block structures.

8. The fundamental objectives of structured programming;

* To increase programmer productivity
* To reduce program testing time
* To reduce program maintenance time and effort
* To increase program clarity by reducing complexity.

9. Logical pattern that characterize structured programming;

* Sequence: refers to an ordered exeuction of statements. It has one entry and exit points.
* Selection: it uses conditions and one of number of statements is executed depending on the state of the program.

Repetition: is executed until the program reaches a certain state or operations have been applied to every element of a collection.

13. Difference between object oriented programming and structured;

 Structured programming

* Difficult to modify structured programs
* Main method communicates with functions by calling those functions in the main program
* There are no access specifers
* Data is not secured

 Object oriented

* Easier to modify objects oriented programs
* Objects communicate with each other by passing messages
* There are no access specifers such as private public and protetcted
* Data is secure.

11.



14. //program to find the circumference of a circle. means that any command won`t be considered for compilation and exeuction.

#include #define pi 3.142 means that it is a preprocessor command that includes standard input and output header file from the C library before compiling a C program.

int main () means that this the function from where exeuction of any C program begins.

printf("the value of the circumference of a circleis %d",cir); means that the command prints the output on the screen.

return 0; this command is terminates C program (main function) and return 0.

} means end of the function.