

**NAME: OGUNLEYE MOFEYISOPE SIMISOLA**

**MATRIC NUMBER: 19/sci01/099**

**DEPARTMENT: COMPUTER SCIENCE 200LEVEL**

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

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

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

**2) The 3 types of programming languages are:**

- **Machine language:** This is the native language directly understood by the computer's central processing unit. This type of computer language is not easy to understand, as it only uses a binary system, an element of notations containing only a series of numbers consisting of one and zero, to produce commands.
- **Assembly language:** It is a language that consists of some symbolic codes, which are easier to remember than machine codes. It makes use of an assembler to translate codes into machine language.
- **High level language:** They are user friendly languages which are similar to English with vocabulary of words and symbols. They are easier to learn and require less time to write. They are also problem oriented rather than machine based.

**3) Some features of programming language are:**

- It must have a vocabulary that consists of letters of the alphabet.
- It must have a language structure, which consists of keywords, expressions, and statements.
- It may require a translator before it can be understood by a computer.
- It must have syntactic rules for forming statements.

**4) Programming methodologies:**

- **Functional Programming:** this is a collection of mathematical functions, each with an input and a result. Interaction and combination of functions is carried out by functional compositions, conditionals, and recursion. Major functional programming languages are Lisp, scheme, Haskell, and ML.
- **Object oriented Programming:** An object oriented program 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.
- **Event driven Programming:** An event driven program is a continuous loop that responds to events that are generated in an unpredictable order. These events originate from user actions on the screen or from other sources. Major event driven programming languages are Java and Visual basic.
- **Concurrent Programming:** This is a collection of cooperating processes, sharing information with each other from time to time but generally operating asynchronously.
- **Procedural Programming:** This is a series of steps, each of which performs a calculation, retrieves input, or produces output. Concepts like assignments, loops and sequences are the building rocks of procedural programming.
- **Logic Programming:** This is a collection of logical declarations about what outcome a function should accomplish rather than how that outcome should be accomplished. It is a natural vehicle for expressing non-determinism, since the solutions to many problems are often not unique but manifold.

5) The major 5 stages of developing a program for Johnson LTD are:

- **Problem analysis:** This is where the clear statement of the problem is stated. The programmer must make sure he understands the problem and how to solve it. He must know what is expected of the problem, i.e., what the problem should do, the nature of the output and the input to consider so as get the output. He must also understand the ways of solving the problem and the relationship between the input and the expected output.
- **Design:** The planning of the solution to the problem in the first stage takes place in this stage. The planning consists of the process of finding a logical sequence of precise steps that solve the problem. Such a sequence of steps

is called an ALGORITHM. Every detail including obvious steps should appear in the algorithm. Planning may also involve using representative data to test the logic of the algorithm by hand to ensure that it is correct.

- Coding: Translation of the algorithm in stage two into a programming language takes place here. The process for writing the program is called CODING. The programmer uses the algorithm devised in the design stage along with the choice of programming language he got from stage 3.
- Testing and debugging: The process involves the location and removal of error in the program if any. Testing the process of checking if the program is working as expected and finding errors in the program while debugging is the process of correcting errors that are found.
- Documentation: This is the final stage of program development. It consists of organizing all the material that describes the program. The documentation of the program is intended to allow another person or the programmer at a later date, to understand the program. Documentation might also consist of a detailed description of what the program does and how to use the program.

6) When writing a good program you must put some principles into consideration:

- File naming: Files should be organized into directories in a module wise fashion instead of having a monolithic structure where all source code files and header files are in a single directory. This should be part of the design process.
- Naming conventions: It is very important to give meaningful names to all your constructs. The name of a class should communicate its purpose. Class name should start with an uppercase letter.
- Formatting and indentations: The lines within the code should be clearly organized in a way that it will be easy to read and understand even for the writer.
- Comments and documentation: Introducing comments and proper explanations 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.

- Classes: Ensure all the classes in your application have a default constructor, copy constructor and overload operator.
- Functions: A function should normally do one job and do it well. Avoid generic functions with lots of conditional branches to do everything. Make functions simple and small. The ideal size of functions is around 35 – 40 lines.
- Using STL: Use Standard Template Library instead of creating your own container data structures.
- Minimizing bugs by testing: Testing is an integral part of software development. Tests helps us not only in making sure that what we have written is correct, but also find out if someone breaks the code later.

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

8) Objectives of structured programming:

- To increase the programmers productivity
- To reduce program testing time.
- To increase program clarity by reducing complexity.
- To reduce program maintenance time and effort.

9) Sequence: This refers to an ordered execution of statements. It has one entry and exit point.

- 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 if...then....else....endif, switch, or case.
- Repetition: In repetition a statement is executed until the program reaches a certain state, or operations have been applied to every element of a collection. This is usually expressed with keywords such as while, repeat, for or do...until. Often it is recommended that each loop should have only one entry point.

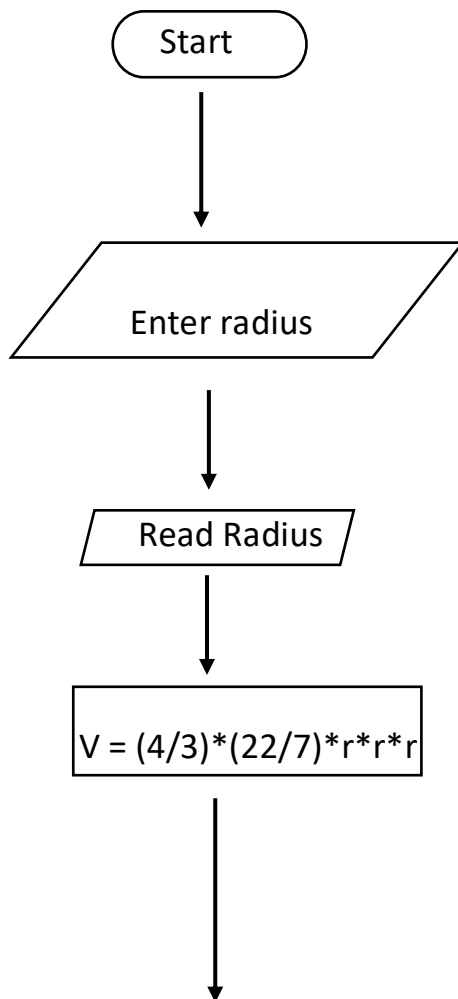
10) The method to be used in solving the problem is through the use of algorithms. The properties of the algorithm to be used are:

- Input: An algorithm has input values from a specified set.
- Output: From each set of input values an algorithm produces an output values from a set of specific set.
- Definiteness: The steps of an algorithm must be defined precisely.

Correctness: An algorithm should produce the correct output values for each set of input values.

- Finiteness: An algorithm should produce the desired output after a finite number of steps for any input in the set.
- Effectiveness: It must be possible to perform each step of an algorithm exactly and in a finite amount of time.
- Generality: The procedure should be applicable for all problems of the desired form, not just for a particular set of input values.

11.



Algorithm:

Step 1: Start

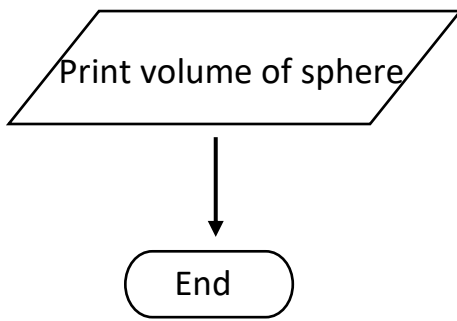
Step 2: Enter radius = 23

Step 3: Read radius

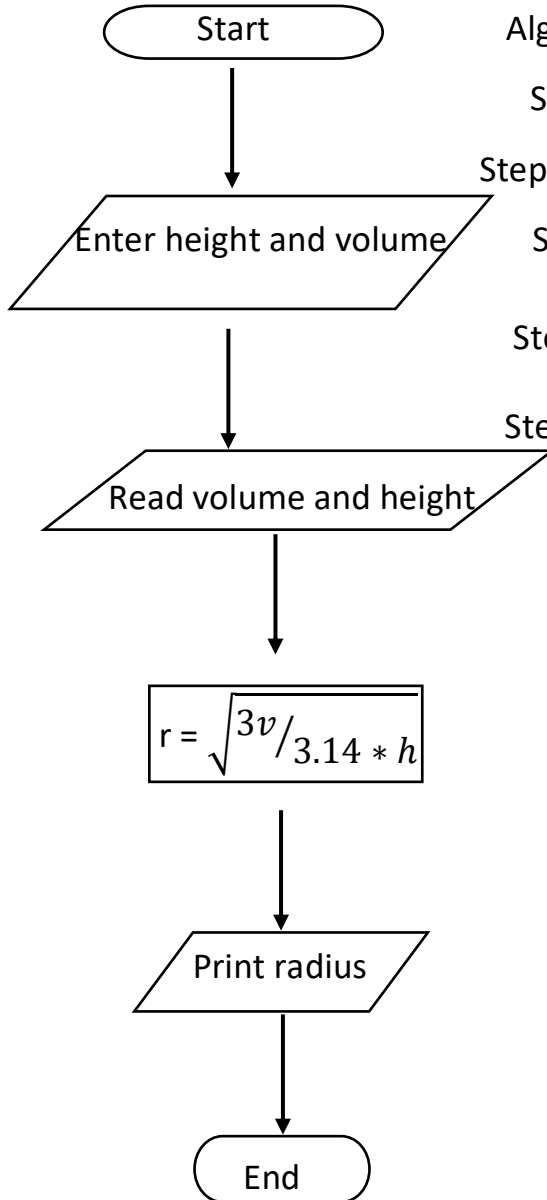
Step 4:  $V = (4/3) * (22/7) * r * r * r$

Step 5: Print volume of sphere(V)

Step 6: Stop



12.



Algorithm:

Step 1: Start

Step 2: Enter volume = 200cm<sup>3</sup> and height =18

Step 3: Read volume and height

Step 4:  $r = \sqrt{3v / 3.14 * h}$

Step 5: Print radius (r)

Step 6: Stop

13) Object oriented programming languages are designed to focus on data while structure oriented programming language focuses on process. Object oriented

programming languages follows top-down approach while structure oriented programming language follows bottom up approach.

14)Line 1: `//program to find the circumference of a circle` : This is the DOCUMENTATION SECTION. the statement in the command `“//”` wont be considered for compilation and execution.

Line 2: `#include <stdio.h>` is a preprocessor command that includes standard input output header file (stdio.h) from the C library before compiling a C program. It is the LINK SECTION.

Line 3: `#define pi 3.142`: This is the DEFINITION SECTION. `#define` defines a preprocessor macro. It defines a name(pi) against a value (3.142), so before compiling source code , compiler replaces the name with the value in the source code.

Line 4: `int main()` is the main function from where execution of any C program begins. It is the MAIN FUNCTION.

Line 5: `{` : This indicates the beginning of the main function.

Line 6: `const float rad= 7.5` : The `const` tells the computer that the variable (rad = 7.5) is a read-only variable and that it won't be changed throughout its existence. The `float` is a fundamental data type built into the compiler used for defining numeric values with floating decimal points.

Line 7: `float circ`: This means the answer of the circumference(circ) will be in decimal.

Line 8: `circ= 2*pi*rad`: This is the section where User functions are defined. This statement is defining the formula the program will use to calculate the circumference(circ).

Line 9: `printf(“the value of the circumference of a circle is %d”,circ)`: `printf` is a command that prints the output onto the screen, in this case the `printf` prints everything in the quotation and inserts the output where the `%d` is. `%d` in the statement is used to display the value of an integer variable. Also, `circ` in the statement indicates the variable to be displayed in place of `%d`.

Line 10: `return 0;` This command terminates the C program (main function) and returns 0.

Line 11: `}` This indicates the end of the main function.