SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
COIMBATORE – 35

DEPARTMENT OF COMPUTER SIENCE AND ENGINEERING (UG & PG)

Third Year Computer Science and Engineering, 3rd Semester

2 Marks Question and Answer

Subject Code &Name: 23CSB201 & OBJECT ORIENTED PROGRAMMING

Prepared by: Dr.S.R.JANANI, AP/CSE

UNIT I

1. State the characteristics of procedure oriented programming.

   • Emphasis is on algorithm.

   • Large programs are divided into smaller programs called functions.

   • Functions share global data.

   • Data move openly around the system from function to function.

   • Functions transform data from one form to another.

   • Employs top-down approach in program design.

2. What are the features of Object Oriented Programming?

   • Emphasis is on data rather than procedure.

   • Programs are divided into objects.

   • Data structures are designed such that they characterize the objects.

   • Functions that operate on the data of an object are tied together.

   • Data is hidden and cannot be accessed by external functions.

   • Objects may communicate with each other through functions.

- New data and functions can easily be added whenever necessary.
- Follows bottom-up approach.

3. Distinguish between Procedure Oriented Programming and Object Oriented Programming.

   Procedure Oriented Programming Object Oriented Programming

   - Emphasis is on algorithm.
   - Large programs are divided into smaller programs called functions.
   - Functions share global data.
   - Data move openly around the system from function to function.
   - Employs top-down approach in program design.
   - Emphasis is on data rather than procedure.
   - Programs are divided into objects.
   - Functions that operate on the data of an object are tied together.
   - Data is hidden and cannot be accessed by external functions.
   - Follows bottom-up approach.

4. Define Object Oriented Programming (OOP).

Object Oriented Programming is an approach that provides a way of modularizing

programs by creating partitioned memory area for both data and functions that can be

used as templates for creating copies of such modules on demand.

5. List out the basic concepts of Object Oriented Programming.

   - Objects

- Classes
- Data Abstraction and Encapsulation
- Inheritance
- Polymorphism
- Dynamic Binding
- Message Passing

6. Define Objects.

   Objects are the basic run time entities in an object oriented system. They are instance of a class. They may represent a person, a place etc that a program has to handle. They may also represent user-defined data. They contain both data and code.

7. Define Class.

   Class is a collection of objects of similar data types. Class is a user-defined data type. The entire set of data and code of an object can be made a user defined type through a class.

8. Define Encapsulation and Data Hiding.

   The wrapping up of data and functions into a single unit is known as data encapsulation. Here the data is not accessible to the outside world. The insulation of data from direct access by the program is called data hiding or information hiding.

9. Define Data Abstraction.

   Abstraction refers to the act of representing the essential features without including the background details or explanations.

10. Define data members and member functions.

The attributes in the objects are known as data members because they hold the information. The functions that operate on these data are known as methods or member functions.

11. State Inheritance.

    Inheritance is the process by which objects of one class acquire the properties of objects of another class. It supports the concept of hierarchical classification and provides the idea of reusability. The class which is inherited is known as the base or super class and class which is newly derived is known as the derived or sub class.

12. State Polymorphism.

    Polymorphism is an important concept of OOPs. Polymorphism means one name, multiple forms. It is the ability of a function or operator to take more than one form at different instances.

13. List and define the two types of Polymorphism.

    • Operator Overloading – The process of making an operator to exhibit different behaviors at different instances.

    • Function Overloading – Using a single function name to perform different types of tasks. The same function name can be used to handle different number and different types of arguments.

14. State Dynamic Binding.

Binding refers to the linking of procedure call to the code to be executed in response to the call. Dynamic Binding or Late Binding means that the code associated with a given procedure call is known only at the run-time.

15. Define Message Passing.

Objects communicate between each other by sending and receiving information known as messages. A message to an object is a request for execution of a procedure. Message passing involves specifying the name of the object, the name of the function and the information to be sent.

16. List out some of the benefits of OOP.

    - Eliminate redundant code
    - Saves development time and leads to higher productivity
    - Helps to build secure programs
    - Easy to partition work
    - Small programs can be easily upgraded to large programs
    - Software complexity can easily be managed

17. Define Object Based Programming language.

    Object Based Programming is the style of programming that primarily supports encapsulation and object identity. Languages that support programming with objects are known as Object Based Programming languages. They do not support inheritance and dynamic binding.

18. List out the applications of OOP.

    - Real time systems
    - Simulation and modeling
    - Object oriented databases
    - Hypertext, Hypermedia and expertext
    - AI and expert systems
    - Neural networks and parallel programming
    - Decision support and office automation systems
    - CIM/CAM/CAD systems

19. What is the main purpose of using comments in Java code?

Comments are used to annotate the code with explanations, make it more understandable, and provide documentation for future reference. They do not affect the execution of the code.

20. Explain the concept of data types in Java and give two examples.
Data types in Java define the type of data that a variable can hold. They are divided into primitive types (such as int and char) and reference types. Examples of primitive data types are int (for integers) and double (for floating-point numbers).

21. What is the significance of the main method in Java applications?

The main method is the entry point of a Java application. It is where the program starts execution. The method signature is public static void main(String[] args).

22. Describe the role of command line arguments in Java.
Command line arguments allow users to pass inputs to a Java program when it is executed from the command line. These inputs are received as an array of String objects in the main method.

23. What is Java architecture and how does it facilitate portability?
Java architecture includes the Java Virtual Machine (JVM) and the Java Runtime Environment (JRE). It facilitates portability by allowing Java programs to be compiled into bytecode, which can run on any device with a compatible JVM, regardless of the underlying hardware.

24. How do variables differ from data types in Java?
Variables are storage locations in memory with a name and a type, used to hold data. Data types specify the kind of data that can be stored in a variable, such as int, float, or String.

25. Briefly explain the purpose of environment setup in Java development.
Environment setup involves installing the Java Development Kit (JDK), configuring environment variables, and setting up an Integrated Development Environment (IDE) or text editor to write and run Java programs.

26. What is the evolution of Java and its significance?
Java has evolved from its initial release in 1995 with significant updates such as the introduction of generics, lambda expressions, and modules. These changes have improved the language's performance, security, and ease of development.

27. What does it mean for Java to be a "high-level" language?
    Java is considered a high-level language because it abstracts away most of the complex details of the hardware and low-level operations, allowing developers to write code that is more readable and easier to manage.

28. What is the difference between int and double data types in Java?
    int is a primitive data type used to represent whole numbers, whereas double is a primitive data type used for representing floating-point numbers with decimal points.

29. How do you declare a variable in Java and initialize it with a value? Provide an example.
    A variable in Java is declared by specifying its data type followed by its name, and it can be initialized with a value using the assignment operator. For example: int age = 25;.

30. Explain the term "Java bytecode" and its role in Java architecture.
    Java bytecode is the intermediate representation of Java code compiled by the Java compiler. It is platform-independent and executed by the Java Virtual Machine (JVM), allowing Java programs to run on any device with a compatible JVM.

31. What is the role of the Java Virtual Machine (JVM) in Java programming?
    The JVM executes Java bytecode, manages memory, and provides a runtime environment for Java applications, ensuring that the same bytecode can run on any system with a JVM, thereby achieving portability.
32. What is the purpose of the public keyword in Java?
    The public keyword is an access modifier that makes a class, method, or variable accessible from any other class or package, ensuring broad visibility.

33. How does Java achieve platform independence?
    Java achieves platform independence through the use of Java bytecode, which can be executed on any platform that has a Java Virtual Machine (JVM) installed, thus allowing the same bytecode to run on different systems.

34. Define what is meant by "object-oriented programming" in the context of Java.
    Object-oriented programming (OOP) in Java is a paradigm that organizes software design around objects and classes, which encapsulate data and behaviors. It emphasizes principles like inheritance, encapsulation, polymorphism, and abstraction.

35. What is the significance of the static keyword in Java?

The static keyword denotes that a member (variable or method) belongs to the class rather than instances of the class. It allows the member to be accessed without creating an instance of the class.

36. What is the purpose of environment variables in Java development?
Environment variables in Java development configure the system's path to the JDK, set the location of libraries, and control various runtime parameters, ensuring that Java tools and applications run correctly.

37. How does Java's type system help prevent errors in programming?
Java’s type system enforces strict type checking at compile-time, which helps catch type-related errors early, ensuring that variables are used in ways consistent with their declared types and reducing runtime errors.

38. Describe how Java's comment styles (//, /* ... */, and /** ... */) differ from each other.
// is used for single-line comments, /* ... */ for multi-line comments, and /** ... */ for documentation comments (Javadoc) which can be used to generate HTML documentation for classes and methods.