11. Using above created class, Write in brief abstraction and encapsulation

- Definition of Abstraction

Abstraction is an OOP concept that focuses only on relevant data of an object. It hides the

background details and emphasizes the essential data points for reducing the complexity and

increase efficiency. It generally retains only information which is most relevant for that specific

process. Abstraction method mainly focusses on the idea instead of actual functioning.

Definition of Encapsulation

Encapsulation is a method of making a complex system easier to handle for end users. The user

need not worry about internal details and complexities of the system.

Encapsulation is a process of wrapping the data and the code, that operate on the data into a

single entity. You can assume it as a protective wrapper that stops random access of code defined

outside that wrapper.

12. Explain difference among class and object?

- A class is a template for creating objects in a program, whereas the

object is an instance of a class. •

• A class is a logical entity, while an object is a physical entity.

A class does not allocate memory space; on the other hand, an object

allocates memory space. • You can declare a class only once, but you can create more than one

object using a class. •

• Classes can’t be manipulated, while objects can be manipulated. • Classes don’t have any values, whereas objects have their own values.

You can create a class using “class” keyword, while hand you can create

an object using “new” keyword in Java.

13. Define access modifiers?

- Access modifiers (or access specifiers) are keywords in object-oriented languages

that set the accessibility of classes, methods, and other members. Access modifiers are a specific part of programming language syntax used to facilitate the

encapsulation of components.

14. Explain an object? Create an object of above class. - An object is a real-world entity. An object is a runtime entity. The object is an entity which has fff Page 1.2

An object is a real-world entity. An object is a runtime entity. The object is an entity which has

state and behavior. The object is an instance of a class. A class is a group of objects which have

common properties. It is a template or blueprint from which objects are created.

15. Give real life examples of object. - An entity with some state and behaviour is referred to as an object. Objects can be tangible or

intangible. Object Examples : pen, car, bike, table, chair, mobile, etc. The

characteristics an object defines include state, behavior, and identity.

16. Explain a Constructor. -

In Java, a constructor is a block of codes similar to the method. It is called when an

instance of the class is created. At the time of calling constructor, memory for the

object is allocated in the memory.

It is a special type of method which is used to initialize the object.

Every time an object is created using the new() keyword, at least one constructor is

called.

It calls a default constructor if there is no constructor available in the class. In such

case, Java compiler provides a default constructor by default.

a. Constructor name must be the same as its class name

b. A Constructor must have no explicit return type

c. A Java constructor cannot be abstract, static, final, and synchronized

17. Define the various types of constructors?

- There are two types of constructors in Java:

Default constructor (no-arg constructor)

Parameterized constructor

Java Default Constructor

A constructor is called "Default Constructor" when it doesn't have any parameter.

Java Parameterized Constructor

A constructor which has a specific number of parameters is called a parameterized constructor.

Why use the parameterized constructor?

The parameterized constructor is used to provide different values to distinct objects. However,

you can provide the same values also.

18. Whether static method can use non-static members?

- Restrictions in Static Methods:

Non-static data members or non-static methods cannot be used by static methods, and static

methods cannot call non-static methods directly.

In a static environment, this and super aren’t allowed to be used.

19. Explain Destructor?

- Destructor is an instance member function which is invoked automatically whenever an object is

going to be destroyed. Meaning, a destructor is the last function that is going to be called before

an object is destroyed. Destructor is also a special member function like constructor

20. Explain an Inline function?

- An inline function is one for which the compiler copies the code from the function definition

directly into the code of the calling function rather than creating a separate set of instructions in

memory. This eliminates call-linkage overhead and can expose significant optimization

opportunity