oop

Subject: Object Oriented Programming Course Code: CMP-2123

Objective Part

Q.NO.1 Write short answers of the following questions in 3-4 lines only (12*2=24)

1. Define concrete class?

Concreteclass is any such class which has implementation of all its inherited members.

2. Explain garbage collection in Java.

The Java Virtual Machine automatically reclaims all memory allocated to unreferenced objects for future use. In other words, if an object is no longer referenced and accessible the memory allocated to that object is freed and made available for the creation of other objects. This clean up process is called garbage collection.

3. Describe any two access modifiers supported by java.

There are different types of access modifiers, default, public, private, protected.

Private

Private data members of a class are only accessible in the class in which they are declared.

Public

The public data members are accessible from anywhere in the program.

Protected

A protected variable or method is visible to its defining class and all its subclasses as well as any other classes in the same packages.

4. What is ad-hoc polymorphism? give an example

The ad hoc polymorphism is a technique used to define the same method with different implementations and different arguments. In a java programming language, ad hoc polymorphism carried out with a method overloading concept. In ad hoc polymorphism the method binding happens at the time of compilation. Ad hoc polymorphism is also known as compile-time polymorphism.

The most common example of ad-hoc polymorphism is overloading, which associates a single function symbol with many implementations; the compiler (or the runtime system, depending on whether overloading resolution is static or dynamic) chooses an appropriate implementation for each application of the function, based on the types of the arguments.

5. Define Final methods and classes?

Final methods and **final classes** are also similar. A **class** declared as final can not be extend while a **method** declared as final cannot be overridden in its subclasses. A **method** or a **class** is declared to be final using the final keyword.

11. What is wrapper class? Give an example.

In object-oriented programming, a wrapper class is a class that encapsulates types, so that those types can be used to create object instances and methods in another class that need those types. Example: wrapper classes are used for converting primitive data types into objects, like int to integer etc.

Primitive	Wrapper class
int	Integer
long	Long
float	Float
double	Double

12. What is different between downcasting and upcasting.

Down casting means casting an object to a derived or more specialized type. Upcasting allows a base type reference to refer to an object of a derived type. Down casting is a casting to a subtype. Upcasting is a casting to super type. Down casting involves a type check and can throw a ClassCastException. Upcasting is always allowed.

13. Write down five common examples of exception.

ClassNotFoundException

IOException

FileNotFoundException

ArithmeticException

NullPointException

IndexOutOfbBoundException

EOF (End of File) Exception

14. Differentiate between encapsulation and information hiding.

Encapsulation is the grouping of related ideas into one unit, which can thereafter be referred to by a single name.

Information hiding is the principle that hides information details of an object or function. The hiding of these details results in an abstraction which reduces the external complexity and makes the object or function easier to use.

15. Differentiate between getter and setter method.

Getter method

A method that returns the value of some private variables is called a getter method.

keyword at a time. With throws keyword both checked and unchecked can be declared and propagate using throws

23. What is Encapsulation?

Encapsulation is the grouping of related ideas into one unit, which can thereafter be referred to by a single name.

24. Differentiate an Interface and abstract class?

The main difference is methods of a java interface are implicitly abstract and cannot have implementations. A java abstract class can have instance methods that implements a default behavior. Variables declared in a java interface is by default final. An abstract class may contain non-final variables.

25. Define Final method and Final class.

Final methods and classes are also similar. A class declared as final cannot be extended while a method declared as final cannot be overridden in its subclasses. A method or a class is declared to be final using the final keyword.

26. What is JVM?

A Java Virtual Machine (JVM) is a virtual machine that enables a computer to run Java program as well as programs written in other languages that are also compiled to Java bytecode. The JVM is detailed by a specification that formally describes what is required in a JVM implementation.

27. Define Interface.

An interface is a reference type in java, similar to class collection of abstract method.

28. What are inner classes?

In object-oriented programming an inner class or nested class is a class declared entirely within the body of another class or interface. It is distinguished from a subclass.

29. Why main () method is static?

Java main () method is always static, so that compiler can call it without the creation of an object or before the creation of an object of the class. In an java program main () method is starting point from where compiler starts program execution. So, the compiler needs to call the main () method.

30. Why String Buffer is called mutable?

String Buffer are called mutable because whenever we perform a modification on their objects their static gets changed. And they were created as mutable because of the immutable nature of string class.

6. Early and late binding are used for which type of methods?

The early binding just means that the target method is found at compile time while in late binding the target method is looked up at run time. Most script languages use late binding and compiled languages use early binding.

7. What is difference b/w boxing and Unboxing?

Boxing

Boxing is the process of converting a value type to the type object. Boxing is implicit conversion.

Unboxing

In Unboxing an object stored on heap memory is copied to a value type stored on stack memory. Unboxing is explicit conversion.

8. Describe checked and unchecked exceptions?

Checked Exception

An exception that is not unchecked is called a checked exception. A checked exception is one from which a method can reasonably be accepted to recover.

Unchecked Exception

An unchecked exception such as an out of bounds array index, is one that usually cannot be handled during runtime.

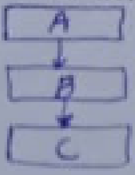
9. Describe run time polymorphism.

Runtime polymorphism is a process in which a call to an overridden method is resolved at runtime rather than compile-time. In this process, an overridden method is called through the reference variable of the super class.

10. Differentiate between multilevel and multiple inheritance.

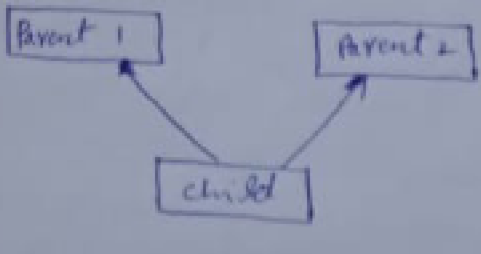
Multi~~level~~ ~~ple~~ level Inheritance

A type of inheritance in which a class is derived from another derived class is called multi~~p~~level inheritance



Multi~~level~~ple Inheritance

A type of inheritance in which a child class is derived from a multiple parent class is known as multiple inheritance.



Setter method

A method that assigns or alters the value of one of the instance variables is called a setter method.

16. Describe static method. How can a static method be invoked?

In Java a static method is a method that belongs to a class rather than an instance of a class. The method is accessible to every instance of a class but method defined in an instance are only able to be accessed by that member of a class. A static method cannot invoke an instance method accept via an object.

17.What happens if no catch handler matches the type of a thrown object?

This cause the search for a match to continue in the next enclosing try statement. If there is a finally block, it will be executed before the exception goes to the next enclosing try statement.

18. What is abstract class? Can a class be declared as abstract with all concrete method?

A class which is declared with the abstract keyword is known as an abstract class in java. They cannot be instantiated so if you have an abstract class with concrete methods then it can be sub classed and subclass can then be instantiated.

19. Define default and No-Argument Constructor?

If the class being declared is the primordial class object, then the default constructor has an empty body. Otherwise the default constructor simply invokes the super class instructor with no arguments. It is the constructor that is added by the compiler if you have not defined a constructor.

20. Define JRE i.e. Java Runtime Environment?

Java Runtime Environment (JRE) is a set of software tools for development of java applications. It combines the java virtual machine (JVM), platform core classes and supporting libraries. JRE is part of the java development kit (JDK), but can be downloaded separately.

21. Why string class is considered immutable?

The string is Immutable in Java because string objects cashed in string pool. Another reason of why string class is immutable could die due to HashMap. Since Strings are very popular as HashMap key, it's important for them to be immutable so that they can retrieve the value object which was stored in HashMap.

22. what is difference between throw and throws?

Throw

Throw is a keyword which is used to throw an exception explicitly in the program inside a function. Internally throw is implemented as it is allowed to throw only single exception at a time. With throw keyword we can propagate only unchecked exception.

Throws

Throws is a keyword used in the method signature used to declare an exception which might get thrown by the function while executing the code. We can declare multiple exceptions with throws

31. What is polymorphism or dynamic method dispatch?

Dynamic method dispatch allow java to support overriding of methods which is central for run-time polymorphism. It allows a class to specify methods that will be common to all of its derivatives, while allowing subclasses to define the specific implementation of some or all of those methods.

32. Differences between this and super keyword.

This keyword

Represents a current instance of a class. Differentiate the instance variables from local variables if they have same names, within a constructor or a method.

Super keyword

Super represent the current instance of the parent class. It is used to differentiate the members of superclass from the members of subclass, if they have same names.

33. Differentiate b/w static and dynamic binding?

Static binding

It is resolved at a compile time. Static binding use type of the class and fields. Overloading is an example of static binding. Private, final and static methods and variables uses static binding.

Dynamic binding

It is resolved at run time. Dynamic binding uses object to resolve binding. Method overriding is the example of dynamic binding. Virtual methods use dynamic binding.

34. Explain the importance of information hiding in OOP?

Information hiding is the principle of segregation of the design decisions in a computer program that are most likely to change, thus protecting other parts of the program from extensive modification if the design decision is changed. That is why it is important.