Date __/__/____

Saathi

"Few Questions"

(1) what is Inheritance in Java?

⇒ Inheritance is the process where child class acquiring the properties and behaviours of the parent class by using extend keyword.

(1) It always speaks about code reusability.
It implements DRY principle (Do not repeat yourself)

(2) By using inheritance the productivity of code can be improved and lines of code can be reduced in the application.

we cannot inherits the private properties of parent class because of to preserve the concept of encapsulation.

(2) what is superclass and subclass?

⇒ A class from where a subclass inherits features is called superclass. It is also called parent class or base class.

A class that inherits all the members (fields, methods) from another class is called subclass. It is also called derived class or child class.

(3) How is Inheritance Implemented/achieved in Java?

⇒ Inheritance can be implemented or achieved by using two keywords:-

(1) Extends ⇒ Extends keyword is used for developing the inheritance between two classes and two interfaces.

(2) Implements ⇒ Implements keyword is used for developing the inheritance between a class and interface.

(4) what is polymorphism?

polymorphism means when the behaviour of same object is different in different contexts. It is mainly divided into two types:-

(1) static polymorphism ⇒

⇒ overloading. If polymorphism exist at compilation time then it is called compile time or static polymorphism.

(2) Dynamic polymorphism ⇒

If polymorphism exist at runtime then it is called runtime or dynamic polymorphism.

Ex ⇒ overridding.

To achieve polymorphism we create parent class reference for child class object.

(5) Differentiate between method overloading and method overridding

⇒ Method overloading ⇒

(1) It implements compile time polymorphism.

(2) The process of extending existing method functionality with new functionality is called method overloading.

(3) In case of method overloading, different method signatures must be provided to the methods.

(4) with or without inheritance we can perform method-overloading. and the error can be caught at compile time. And It occurs between the methods in same class.

⇒ method overridding ⇒

(1) It implements run time polymorphism.

(2) The process of replacing existing method functionality with new functionality is called method overridding.

(3) In case of methodoverridding, same method prototypes must be provided to the methods.

Date ___/___/_____

Saathi

(4) With inheritance only we can perform methodoverriding and the error will ~~be~~ visible at runtime And. It occurs between superclass and subclass.

(6) what is an abstraction explained with an example?
Abstraction means hiding the internal details and showing the essential things to user. we can achieve abstraction by using abstract keyword. abstract keyword can be applied with method and class. when we override a method from parent class in child class then in parent class we should declare the method abstract.

Abstract method ⇒
The method is declared without any implementation it is called abstract method and if any class there is in abstract method then we must be declared the class abstract.

Abstract class ⇒
A class which have atleast one abstract method then we must be declared the class abstract it is called abstract class.
It is not possible to create a object of abstract class but it is possible we create abstract class reference for other class object.

(7) what is difference between abstract method and final method in java?

⇒ The abstract method is incomplete while the final method is regarded as complete. The only way to use an abstract method by overriding it but we cannot override final method in java. The combination of final and abstract is illegal.

Page No.

Date ___/___/___

(8) what is the final class in java?

⇒ If we declare a class final then we cannot inherit the class. we can create a class as final only if it is complete in nature which means it must not be an abstract class. In java, all the wrapper classes are final classes like string, integer etc.

(9) Differentiate between abstraction and encapsulation?

⇒ Abstraction	Encapsulation
(1) Abstraction hides the unnecessary details and showing the essential information	Encapsulation hides the code and data into a single unit so that the data can be protected from outside world
(2) Abstraction solves an issue at design level	Encapsulation solves an issue at implementation level
(3) Abstraction focuses on external lookout	Encapsulation focuses on internal working.
(4) Abstraction can be implemented by using abstract classes and interfaces	Encapsulation can be implemented by using access modifiers (Public, protected, private)
(5) In Abstraction, we use abstract classes and interfaces to hide code complexities	In encapsulation, we use getters and setters method to hide the data.
(6) In Abstraction, the objects are encapsulated that helps to perform abstraction.	In encapsulation the object need not to abstract that result in encapsulation.

Saathi

Date ___ / ___ / ___

(10) Differentiate between Runtime and compile time polymorphism

Compile time polymorphism	Runtime polymorphism
(1) Compile time polymorphism is less flexible as all the things execute at compile time	Runtime polymorphism is more flexible as all the things execute at runtime.
(2) In compile time polymorphism, the call is resolved by compiler.	In runtime polymorphism, the call is not resolved by compiler.
(3) In compile time polymorphism inheritance is not involved	In runtime polymorphism inheritance is involved.
(4) C.T.P also known as static binding, Early binding and overloading as well.	R.T.P also known as Dynamic binding, late binding and overriding as well.
(5) C.T.P provides fast execution because the method that needs to be executed is known early at compile time.	R.T.P provides slow execution as compared to early binding because the method that needs to be executed is known at runtime.
(6) method overloading is compile time polymorphism where two or more methods have same name but different parameters and different datatypes.	method overriding is runtime polymorphism having same method with same parameters but associated with compared, different classes.