OOP(JAVA) QUESTIONS

Topics: Array of Object, Inheritance, Abstract class, interface, Exception.

1. Create a class ‘Student’ with three data members which are name, age and address. The constructor of the class assigns default values name as “unknown”, age as ‘0’ and address as “not available”. It has two members with the same name ‘setInfo’. First method has two parameters for name and age and assigns the same whereas the second method takes has three parameters which are assigned to name, age and address respectively. Print the name, age and address of 10 students. (Use array of objects).
2. Define an Employee class with suitable attributes having getSalary() method, which returns salary withdrawn by a particular employee. Write a class Manager which extends a class Employee, override the getSalary() method, which will return salary of manager by adding traveling _allowance, house rent allowance etc.
3. Write a program to explain the use of constructor and override Method in multilevel inheritance up to three level such that:
   1. A is parent of B
   2. B is parent of C
   3. C is parent of D
   4. D is parent of E

- Include no parameter and two parameters constructor, and display() method in each class.
- Create object with no parameter and two parameters for child class, such that creation of object with no parameters call all possible no parameter constructor and creation of object with two parameters call all one parameters constructor. Call the child display() method in chain.

4. Write a Java program to create a base class Animal with methods eat () and. Create three subclasses: Lion, Tiger, and Panther. Override the eat () method in each subclass to describe what each animal eats. In addition, override the sound () method to make a specific sound for each animal.
5. Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.
6. Create an abstract class 'Bank' with an abstract method 'getBalance'. $100, $150 and $200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.
7. An abstract class has a construtor which prints "This is constructor of abstract class", an abstract method named 'a_method' and a non-abstract method which prints "This is a normal method of abstract class". A class 'SubClass' inherits the abstract class and has a method named 'a_method' which prints "This is abstract method". Now create an object of 'SubClass' and call the abstract method and the non-abstract method.

8. Create an abstract class 'Animals' with two abstract methods 'cats' and 'dogs'. Now create a class 'Cats' with a method 'cats' which prints "Cat's meow" and a class 'Dogs' with a method 'dogs' which prints "Dogs bark", both inheriting the class 'Animals'. Now create an object for each of the subclasses and call their respective methods.

9. Write a Java program to create an abstract class BankAccount with instance variable accountnumber and balance, use constructor for 1st value. Also have an abstract methods deposits () and withdraws () and non-abstract method accountnumber and getbalance just return account number and balance respectively. Abstract class has a method setbalance to set the balance. Create subclasses: SavingsAccount and CurrentAccount that extend the BankAccount class and implement the respective methods to handle deposits () and withdrawals () for each account type.

10. Write a java program which creates an interface IterF1 having 2 methods add () and sub (). Create a class which overloads the given methods for addition and subtraction of two numbers respectively.

11. Write a Java programming to create a banking system with three classes - Bank, Account, SavingsAccount, and CurrentAccount. The bank should have a list of accounts and methods for adding them. Accounts should be an interface with methods to deposit, withdraw, calculate interest, and view balances. SavingsAccount and CurrentAccount should implement the Account interface and have their own unique methods.

12. All statement defines in one package.

a. Create an abstract class pen with methods write () and refill () as abstract methods
b. Use the pen class from Q1 to create a concrete class fountain pen with additional method change Nib ()
c. Create a class monkey with jump ( ) and bite ( ) methods Create a class human whichinherits this monkey class and implements basicanimal interface with eat ( ) and sleep methods
d. Create a class telephone with ( ) , lift ( ) and disconnected ( ) methods as abstract methods create another class smart telephone and demonstrate polymorphism
e. Demonstrate polymorphism using using monkey class from point c.
f. Create an interface TVremote and use it to inherit another interface smart TVremote
g. Create a class TV which implements TVremote interface from point f.

Write a Package MCA which has one class Student. Accept student detail through