**CDAC Mumbai PG-DAC August 24**

**Assignment No- 5**

1. Create a base class BankAccount with methods like deposit() and withdraw(). Derive a class SavingsAccount that overrides the withdraw() method to impose a limit on the withdrawal amount. Write a program that demonstrates the use of overridden methods and proper access modifiers & return the details.
2. package org.bankaccount.com;
3. public class BankAccount {
4. private double balance;
6. public BankAccount(double initialBalance) {
7. balance = initialBalance;
8. }
9. public void deposit(double amount) {
10. if(amount > 0) {
11. balance += amount;
12. System.***out***.println("deposited: "+amount);
13. }
14. else {
15. System.***out***.println("deposit amount must be positive");
16. }
18. }
19. public void withDraw(double amount) {
20. if(amount > 0 && amount <= balance) {
21. balance -= amount;
22. System.***out***.println("withdraw: "+amount);
23. }
24. else {
25. System.***out***.println("invalid amount");
26. }
27. }
29. public double getBalance() {
30. return balance;
31. }
32. public String getAccountDetails() {
33. return "balance: "+balance;
34. }
35. }

package org.bankaccount.com;

public class SavingsAccount extends BankAccount{

private double withDrawalLimit;

public SavingsAccount(double initialBalance , double withDrawalLimit) {

super(initialBalance);

this.withDrawalLimit = withDrawalLimit;

}

public void withDraw(double amount){

if(amount>0 && amount <= getBalance() && amount <= withDrawalLimit){

super.withDraw(amount);

System.***out***.println("withdrawal with limit:");

}

else if(amount > withDrawalLimit) {

System.***out***.println("withdrawal amount exceeds thi limit");

}

else {

System.***out***.println("invalid amount");

}

}

public double getWithDrawalLimit() {

return withDrawalLimit;

}

public String getAccountDetails() {

return super.getAccountDetails() + " withdrawal limit :" + withDrawalLimit;

}

}

package org.bankaccount.com;

public class Program {

public static void main(String[] args) {

SavingsAccount amn = new SavingsAccount(50000.00 , 1000.00);

System.***out***.println(amn.getAccountDetails());//account detail

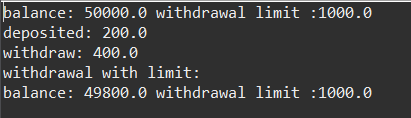
amn.deposit(200.00);//deposit money

amn.withDraw(400.00);//

System.***out***.println(amn.getAccountDetails());

}

}



1. Create a base class Vehicle with attributes like make and year. Provide a constructor in Vehicle to initialize these attributes. Derive a class Car that has an additional attribute model and write a constructor that initializes make, year, and model. Write a program to create a Car object and display its details.
2. package org.vehicle.com;
3. public class Vehicle {
4. private String make;
5. private int year;
6. public Vehicle(String make , int year) {
7. this.make= make;
8. this.year = year;
9. }
10. public String getMake() {
11. return make;
12. }
13. public int getYear() {
14. return year;
15. }
16. }

package org.vehicle.com;

public class Car extends Vehicle {

private String model;

public Car(String make, int year, String model) {

super(make, year);

this.model = model;

}

public String getModel() {

return model;

}

public void displayDetails() {

System.***out***.println("make: "+ getMake());

System.***out***.println("year: "+getYear());

System.***out***.println("model: "+ getModel());

}

}

package org.vehicle.com;

public class Program {

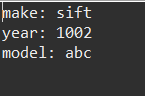
public static void main(String[] args) {

Car c = new Car("sift", 1002 , "abc");

c.displayDetails();

}

}



1. Create a base class Animal with attributes like name, and methods like eat() and sleep(). Create a subclass Dog that inherits from Animal and has an additional method bark(). Write a program to demonstrate the use of inheritance by creating objects of Animal and Dog and calling their methods.
2. package org.animal.com;
3. public class Animal {
4. public Animal() {
6. }
7. public void eat() {
8. System.***out***.println("eat method of animal class");
9. }
10. public void sleep() {
11. System.***out***.println("sleep method animal class");
12. }
13. }

package org.animal.com;

public class Dog extends Animal {

public void bark() {

System.***out***.println("bark method of dog class");

}

}

package org.animal.com;

public class Program {

public static void main(String[] args) {

Animal a = new Animal();

Dog d = new Dog();

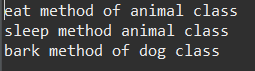
a.eat();

a.sleep();

d.bark();

}

}



1. Build a class Student which contains details about the Student and compile and run its

instance.

package org.student.com;

public class Student {

private String name;

private int age;

private int studentId;

public Student(String name, int age , int studentId) {

this.name = name;

this.age = age;

this.studentId = studentId;

}

public String toString() {

return "Name: "+name+" , age: "+age+" , studenyid: "+studentId;

}

//public String displayInfo() {

// return "Name: "+name+" , age: "+age+" , studenyid: "+studentId;

//}

}

package org.student.com;

public class Problem {

public static void main(String[] args) {

Student s = new Student("mhk", 25, 108);

//System.out.println(s.displayInfo());

System.***out***.println(s.toString());

}

}



1. Write a Java program to create a base class Vehicle with methods startEngine() and stopEngine(). Create two subclasses Car and Motorcycle. Override the startEngine() and stopEngine() methods in each subclass to start and stop the engines differently.
2. package org.vehicle2.com;
3. class Vehicle {
4. // private int start;
5. // private int stop;
6. //
7. // public Vehicle(int start, int stop) {
8. // this.start =start;
9. // this.stop = stop;
10. // }
12. public void startEngine() {
13. System.***out***.println("start engine method of vehicle(superclass) class");
14. }
15. public void stopEngine() {
16. System.***out***.println("stop engine method of vehicle(superclass) class");
17. }
18. }
19. class Car extends Vehicle{
20. *@Override*
21. public void startEngine() {
22. System.***out***.println("start engine method of car(subclass) class");
23. }
24. *@Override*
25. public void stopEngine() {
26. System.***out***.println("stop engine method of car(subclass) class");
27. }
29. }
30. class Motorcycle extends Vehicle{
31. *@Override*
32. public void startEngine() {
33. System.***out***.println("start engine method of motorcyclce(subclass) class");
34. }
35. *@Override*
36. public void stopEngine() {
37. System.***out***.println("stop engine method of motorcycle(subclass) class");
38. }
39. }
41. public class Program {
42. public static void main(String[] args) {
43. Vehicle v = new Vehicle();
44. Vehicle c = new Car();
45. Vehicle m = new Motorcycle();
46. v.startEngine();
47. v.stopEngine();
48. System.***out***.println();
49. c.startEngine();
50. c.stopEngine();
51. System.***out***.println();
52. m.startEngine();
53. m.stopEngine();
54. }
55. }

