**Assignment 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.

**package** org.day10.example.inheritance;

**import** java.util.Scanner;

**class** BankAccount {

**protected** **int** acNo;

**protected** String name;

**protected** **double** balance;

**public** BankAccount() {

**this**.acNo = 0;

**this**.name = "";

**this**.balance=0;

}

// Constructor

**public** BankAccount(**int** acNo, String name, **double** balance) {

**this**.acNo = acNo;

**this**.name = name;

**this**.balance = balance;

}

**public** **void** deposit(**double** amount) {

balance = balance + amount;

}

**public** **void** withdraw(**double** amount) {

**if**(balance > amount) {

balance = balance - amount;

}

**else** {

System.***out***.println("Insufficient Balance !");

}

}

**public** **double** getBalance() {

**return** balance;

}

**public** String toString() {

**return** "\nName : "+ **this**.name+"\nAccount Number : "+ **this**.acNo + "\nBalance : "+ **this**.balance;

}

}

**class** SavingsAccount **extends** BankAccount {

**private** **double** withdrawLimit;

**public** SavingsAccount() {

**this**.withdrawLimit =0;

}

// Parameterised Constructor

**public** SavingsAccount(**int** acNo, String name, **double** balance, **double** withdrawLimit) {

**super**.acNo = acNo;

**super**.name = name;

**super**.balance = balance;

**this**.withdrawLimit = withdrawLimit;

}

**public** **double** getWithdrawLimit() {

**return** withdrawLimit;

}

**public** **void** setWithdrawLimit(**double** withdrawLimit) {

**this**.withdrawLimit = withdrawLimit;

}

@Override

**public** **void** withdraw(**double** amount) {

**if**(amount > withdrawLimit) {

System.***out***.println("Withdrawal limit exceeds! \n Withdrawal Limit : "+ **this**.getWithdrawLimit());

}

**else** {

**super**.withdraw(amount); // calling the Super class method

}

}

**private** Scanner sc = **new** Scanner(System.***in***);

**public** String toString() {

**return** "\nName : "+ **super**.name+"\nAccount Number : "+ **super**.acNo + "\nBalance : "+ **this**.balance;

}

}

**public** **class** ProgramAss1 {

**public** **static** **void** main(String[] args) {

BankAccount acc = **new** BankAccount(1234,"Ganesh", 200000.0);

System.***out***.println("Account Details : "+ acc.toString());

acc.deposit(10500);

System.***out***.println("Account Details : "+ acc.toString());

acc.withdraw(100000);

System.***out***.println("Account Details : "+ acc.toString());

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

SavingsAccount sacc = **new** SavingsAccount(123456, "Ashish", 20000, 2000); // Withdrawal limit 20000

System.***out***.println("Account Details : "+ sacc.toString());

sacc.deposit(10500);

System.***out***.println("Account Details : "+ sacc.toString());

sacc.withdraw(10000);

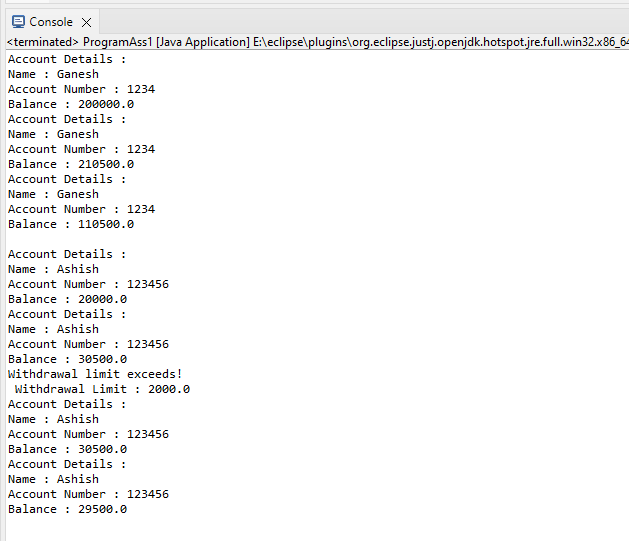
System.***out***.println("Account Details : "+ sacc.toString());

sacc.withdraw(1000);

System.***out***.println("Account Details : "+ sacc.toString());

}

}



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.

**package** org.day10.example.inheritance;

**class** Vehicle{

**protected** String make; // as private will not be accessible in another class taken as protected

**protected** **int** year;

**public** Vehicle() {

**this**.make = "";

**this**.year = 0;

}

**public** Vehicle(String make, **int** year) {

**this**.make = make;

**this**.year = year;

}

**public** String getMake() {

**return** make;

}

**public** **void** setMake(String make) {

**this**.make = make;

}

**public** **int** getYear() {

**return** year;

}

**public** **void** setYear(**int** year) {

**this**.year = year;

}

}

**class** Car **extends** Vehicle{

**private** String model;

**public** Car() {

**super**.make = "";

**super**.year = 0;

**this**.model = "";

}

**public** Car(String model) {

**super**.make = make;

**super**.year = year;

**this**.model = model;

}

**public** String getModel() {

**return** model;

}

**public** **void** setModel(String model) {

**this**.model = model;

}

}

**public** **class** ProgramAss2 {

**public** **static** **void** main(String[] args) {

Car car = **new** Car(); // created instance of class Car which is Sub class

car.setModel("Ford"); // Accessing the members of Super class with instance of sub class

car .setMake("Fiesta");

car.setYear(1990);

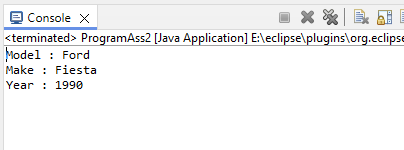
System.***out***.println("Model : " + car.getModel());

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

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

}

}



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.

**package** org.day10.example.inheritance;

**class** Animal{

String name;

**public** **void** eat() {

System.***out***.println("Eating.... | Animal Method");

}

**public** **void** sleep() {

System.***out***.println("Sleeping.... | Animal Method");

}

}

**class** Dog **extends** Animal{

**public** **void** bark() {

System.***out***.println("Barking.... | Dog Method");

}

}

**public** **class** ProgramAss3 {

**public** **static** **void** main(String[] args) {

Dog dog = **new** Dog();

dog.bark();

dog.eat();

dog.sleep();

Animal animal = **new** Animal();

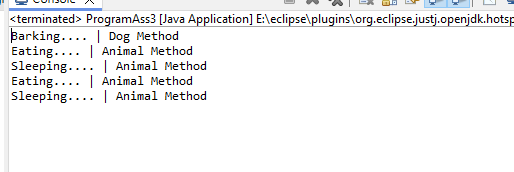
animal.eat();

animal.sleep();

// animal.bark(); Derived class cannot be accessible from the instance of Base class

}

}



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

instance.

**package** org.day10.example.inheritance;

**import** java.util.Scanner;

**class** Student {

**private** **int** id;

**private** String name;

**private** String classSem;

**public** Student() {

**this**.id = 0;

**this**.name = "";

**this**.classSem = "";

}

**public** Student(**int** id, String name, String classSem) {

**this**.id = id;

**this**.name = name;

**this**.classSem = classSem;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getClassSem() {

**return** classSem;

}

**public** **void** setClassSem(String classSem) {

**this**.classSem = classSem;

}

**public** String toString() {

**return** "Class : "+ **this**.classSem + " Id : "+ **this**.id + " Name : " + **this**.name;

}

}

**public** **class** ProgramAss4 {

**private** **static** Scanner *sc* = **new** Scanner(System.***in***);

**public** **static** **void** main(String[] args) {

Student std = **new** Student();

System.***out***.println("Enter Student Class");

std.setClassSem(*sc*.nextLine());

System.***out***.println("Enter Student Id");

std.setId(*sc*.nextInt());

*sc*.nextLine();

System.***out***.println("Enter Student Name");

std.setName(*sc*.nextLine());

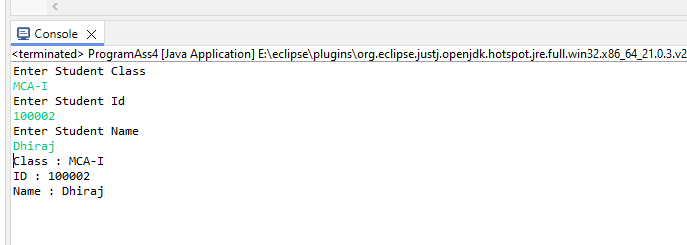
System.***out***.println("Class : "+ std.getClassSem());

System.***out***.println("ID : "+ std.getId());

System.***out***.println("Name : "+ std.getName());

}

}



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.

**package** org.day10.example.inheritance;

**class** VehicleP5{

**public** **void** startEngine() {

System.***out***.println("Start Engine... | class Vehicle");

}

**public** **void** stopEngine() {

System.***out***.println("Stop Engine... | class Vehicle");

}

}

**class** CarP5 **extends** VehicleP5{

@Override

**public** **void** startEngine() {

**super**.startEngine();

System.***out***.println("Start Engine... | class Car");

}

@Override

**public** **void** stopEngine() {

**super**.stopEngine();

System.***out***.println("Stop Engine... | class Car");

}

}

**class** MotorcycleP5 **extends** VehicleP5{

**public** **void** startEngine() {

**super**.startEngine();

System.***out***.println("Start Engine... | Class MotorCycle");

}

**public** **void** stopEngine() {

**super**.stopEngine();

System.***out***.println("Stop Engine... | class MotorCycle");

}

}

**public** **class** ProgrammAss5 {

**public** **static** **void** main(String[] args) {

// Creating instance of class CarP5

CarP5 car = **new** CarP5();

car.startEngine();

car.stopEngine();

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

// Creating instance of class MotorcycleP5

MotorcycleP5 motorcycle = **new** MotorcycleP5();

motorcycle.startEngine();

motorcycle.stopEngine();

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

// Creating instance of Super class which will call only the method of own class

VehicleP5 vehicle = **new** VehicleP5();

vehicle.startEngine();

vehicle.stopEngine();

VehicleP5 veh = **new** CarP5(); // Upcasting

veh.startEngine();

CarP5 car2 = (CarP5) veh; // Downcasting

//

// ClassCastException

VehicleP5 veh5 = **new** VehicleP5(); // not Upcasting

//CarP5 car3 = (CarP5) veh5;

// instanceof - operator which returns boolean value

**if**(veh5 **instanceof** CarP5){

CarP5 car4 = (CarP5) veh5;

}

**else**{

System.***out***.println(" ClassCastException Exception ");

}

}

}

