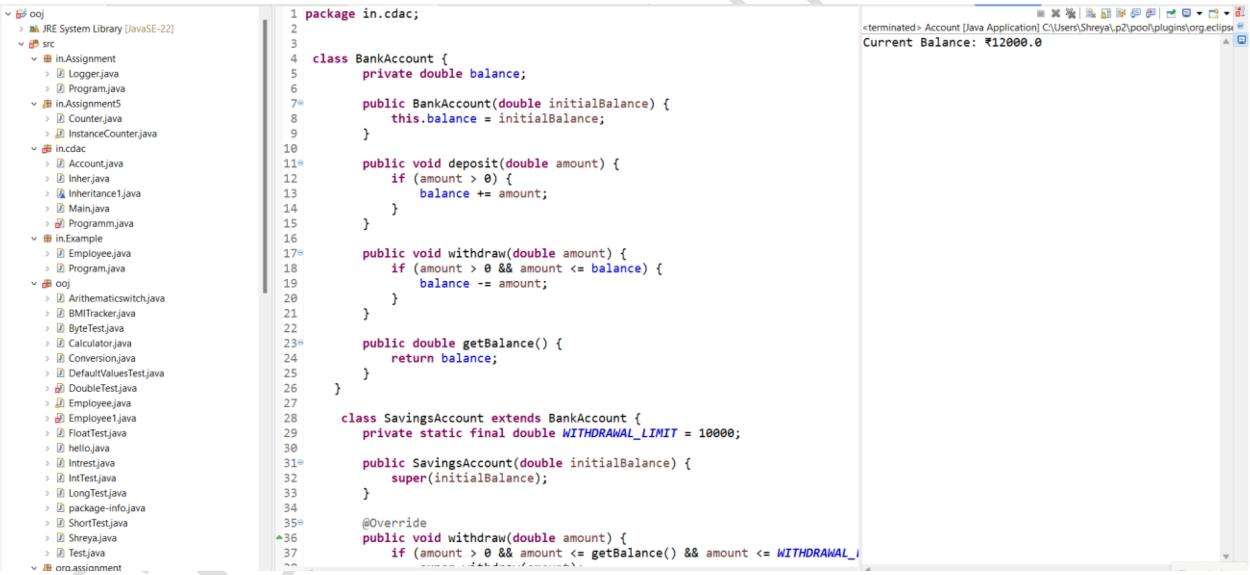
CDAC Mumbai PG-DAC August 24

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.**

```
package in.cdac;

 class BankAccount {
       private double balance;

       public BankAccount(double initialBalance) {
           this.balance = initialBalance;
       }

       public void deposit(double amount) {
           if (amount > 0) {
               balance += amount;
           }
       }

       public void withdraw(double amount) {
           if (amount > 0 && amount <= balance) {
               balance -= amount;
           }
       }

       public double getBalance() {
           return balance;
       }
   }

    class SavingsAccount extends BankAccount {
       private static final double WITHDRAWAL_LIMIT = 10000;

       public SavingsAccount(double initialBalance) {
           super(initialBalance);
       }

       @Override
       public void withdraw(double amount) {
           if (amount > 0 && amount <= getBalance() && amount <= WITHDRAWAL_LIMIT) {
                super.withdraw(amount);
            }
        }
    }



public class Account {
        public static void main(String[] args) {
            SavingsAccount account = new SavingsAccount(15000);

            account.deposit(2000);
            account.withdraw(5000);
            account.withdraw(12000);

            System.out.println("Current Balance: ₹" + account.getBalance());
        }
}
```

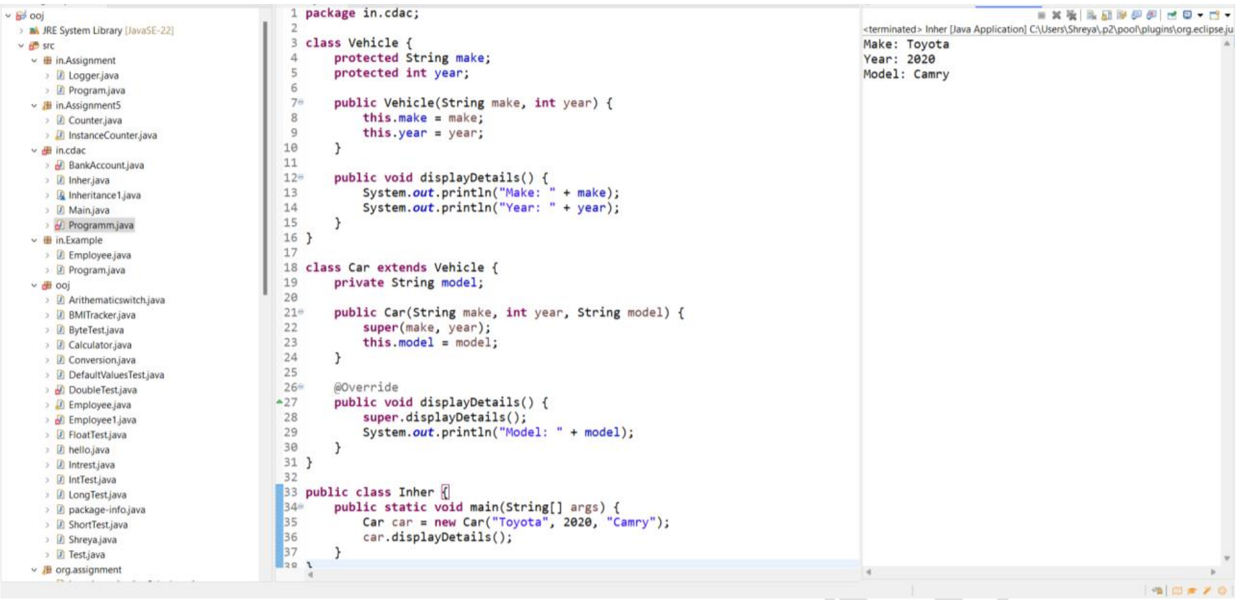


```
package in.cdac;


 class BankAccount {
        private double balance;

        public BankAccount(double initialBalance) {
            this.balance = initialBalance;
        }

        public void deposit(double amount) {
            if (amount > 0) {
                balance += amount;
            }
        }

        public void withdraw(double amount) {
            if (amount > 0 && amount <= balance) {
                balance -= amount;
            }
        }

        public double getBalance() {
            return balance;
        }
    }

     class SavingsAccount extends BankAccount {
        private static final double WITHDRAWAL_LIMIT = 10000;

        public SavingsAccount(double initialBalance) {
            super(initialBalance);
        }

        @Override
        public void withdraw(double amount) {
            if (amount > 0 && amount <= getBalance() && amount <= WITHDRAWAL_
```

```
<terminated> Account [Java Application] C:\Users\Shreya\.p2\pool\plugins\org.eclipse
Current Balance: ₹12000.0
```

2) **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 in.cdac;

class Vehicle {
    protected String make;
    protected int year;

    public Vehicle(String make, int year) {
        this.make = make;
        this.year = year;
    }

    public void displayDetails() {
        System.out.println("Make: " + make);
        System.out.println("Year: " + year);
    }
}

class Car extends Vehicle {
    private String model;

    public Car(String make, int year, String model) {
        super(make, year);
        this.model = model;
    }

    @Override
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Model: " + model);
    }
}

public class Inher {
    public static void main(String[] args) {
        Car car = new Car("Toyota", 2020, "Camry");
        car.displayDetails();
    }
}
```

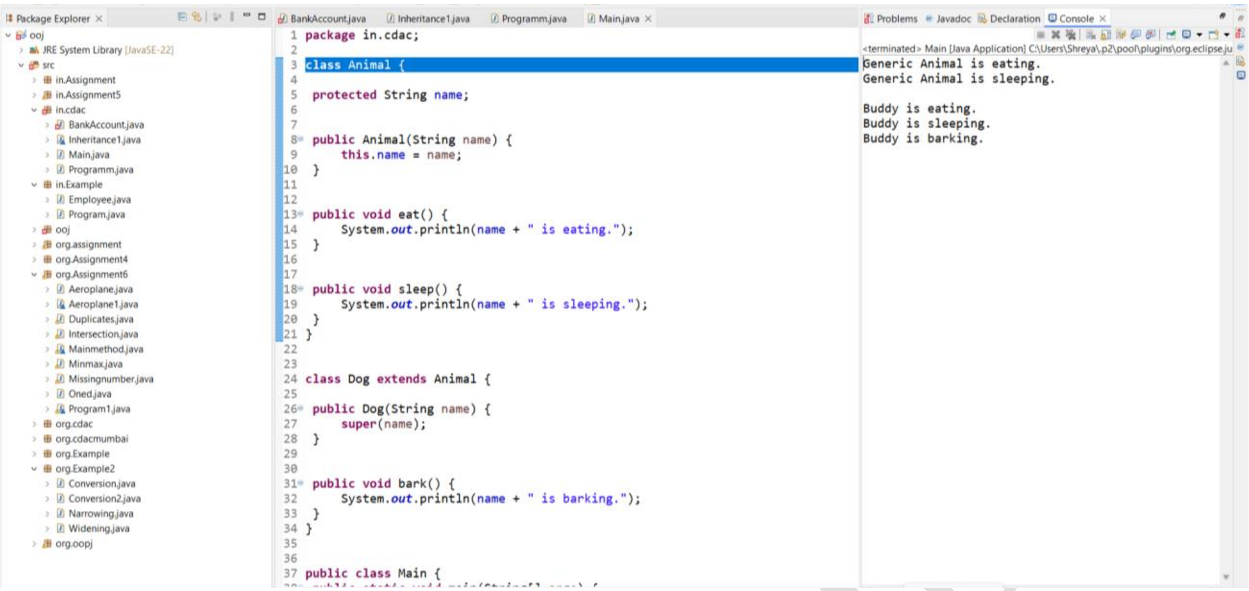


```
package in.cdac;

class Vehicle {
    protected String make;
    protected int year;

    public Vehicle(String make, int year) {
        this.make = make;
        this.year = year;
    }

    public void displayDetails() {
        System.out.println("Make: " + make);
        System.out.println("Year: " + year);
    }
}

class Car extends Vehicle {
    private String model;

    public Car(String make, int year, String model) {
        super(make, year);
        this.model = model;
    }

    @Override
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Model: " + model);
    }
}

public class Inher {
    public static void main(String[] args) {
        Car car = new Car("Toyota", 2020, "Camry");
        car.displayDetails();
    }
}
```

```
<terminated> Inher [Java Application] C:\Users\Shreya\.p2\pool\plugins\org.eclipse.ju
Make: Toyota
Year: 2020
Model: Camry
```

3) **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 in.cdac;

class Animal {

 protected String name;

 public Animal(String name) {

     this.name = name;

 }

 public void eat() {

     System.out.println(name + " is eating.");

 }

 public void sleep() {

     System.out.println(name + " is sleeping.");

 }

}

class Dog extends Animal {
 public Dog(String name) {
     super(name);
 }

 public void bark() {
     System.out.println(name + " is barking.");
 }
}

public class Main {
 public static void main(String[] args) {

     Animal animal = new Animal("Generic Animal");
     animal.eat();
     animal.sleep();

     System.out.println();

     Dog dog = new Dog("Buddy");
     dog.eat();
     dog.sleep();
     dog.bark();
}
      }
```



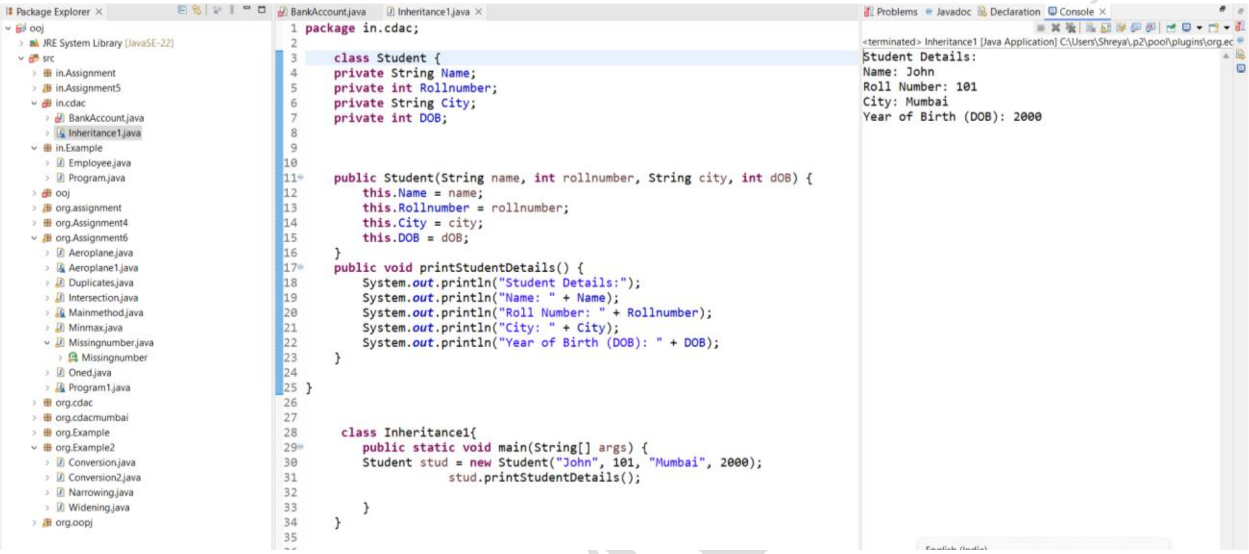
```
package in.cdac;

class Animal {

 protected String name;


 public Animal(String name) {
     this.name = name;
 }


 public void eat() {
     System.out.println(name + " is eating.");
 }


 public void sleep() {
     System.out.println(name + " is sleeping.");
 }
}


class Dog extends Animal {

 public Dog(String name) {
     super(name);
 }


 public void bark() {
     System.out.println(name + " is barking.");
 }
}


public class Main {
```

```
<terminated> Main [Java Application] C:\Users\Shreya\.p2\pool\plugins\org.eclipse.ju
Generic Animal is eating.
Generic Animal is sleeping.

Buddy is eating.
Buddy is sleeping.
Buddy is barking.
```

4) Build a class Student which contains details about the Student and compile and run its instance.

```
package in.cdac;
   class Student {
         private String Name;
         private int Rollnumber;
         private String City;
         private int DOB;

         public Student(String name, int rollnumber, String city, int dOB) {
                  this.Name = name;
                  this.Rollnumber = rollnumber;
                  this.City = city;
                  this.DOB = dOB;
         }
         public void printStudentDetails() {
      System.out.println("Student Details:");
      System.out.println("Name: " + Name);
      System.out.println("Roll Number: " + Rollnumber);
      System.out.println("City: " + City);
      System.out.println("Year of Birth (DOB): " + DOB);
   }

}


          class Inheritance1{
                  public static void main(String[] args) {
```

```
                Student stud = new Student("John", 101, "Mumbai", 2000);
                                stud.printStudentDetails();

                }
        }
```

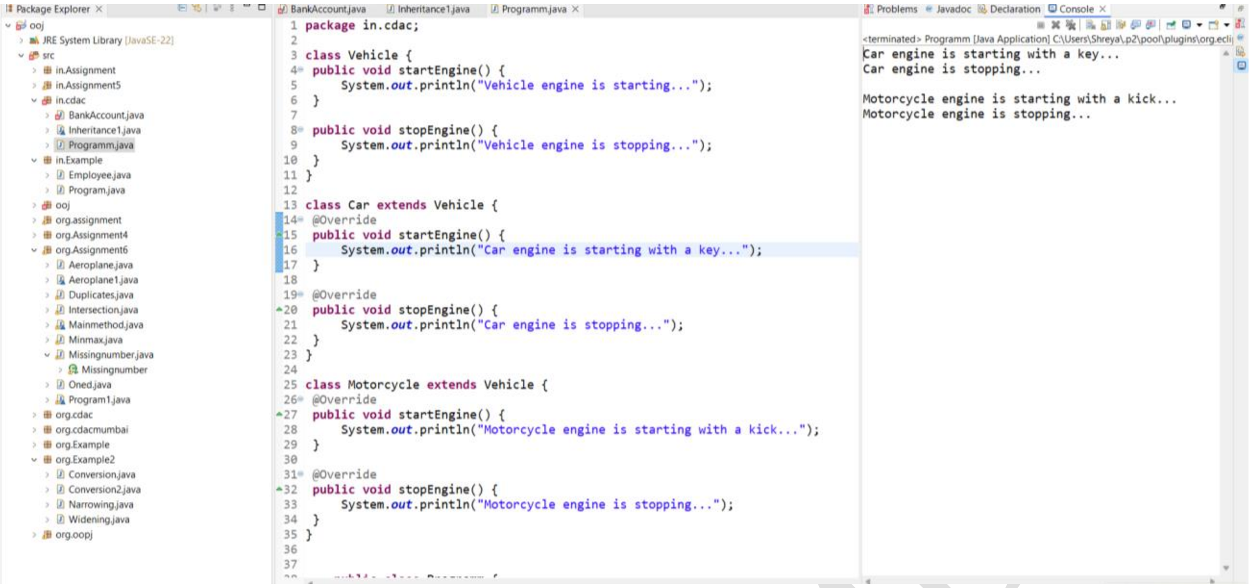


```
package in.cdac;

    class Student {
    private String Name;
    private int Rollnumber;
    private String City;
    private int DOB;



    public Student(String name, int rollnumber, String city, int dOB) {
        this.Name = name;
        this.Rollnumber = rollnumber;
        this.City = city;
        this.DOB = dOB;
    }
    public void printStudentDetails() {
        System.out.println("Student Details:");
        System.out.println("Name: " + Name);
        System.out.println("Roll Number: " + Rollnumber);
        System.out.println("City: " + City);
        System.out.println("Year of Birth (DOB): " + DOB);
    }

}


     class Inheritance1{
        public static void main(String[] args) {
        Student stud = new Student("John", 101, "Mumbai", 2000);
                    stud.printStudentDetails();

        }
    }
```

```
<terminated> Inheritance1 [Java Application] C:\Users\Shreya\.p2\pool\plugins\org.ec
Student Details:
Name: John
Roll Number: 101
City: Mumbai
Year of Birth (DOB): 2000
```

5) 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 in.cdac;
class Vehicle {
 public void startEngine() {
     System.out.println("Vehicle engine is starting...");
 }

 public void stopEngine() {
     System.out.println("Vehicle engine is stopping...");
 }
}

class Car extends Vehicle {
 @Override
 public void startEngine() {
     System.out.println("Car engine is starting with a key...");
 }
 @Override
 public void stopEngine() {
     System.out.println("Car engine is stopping...");
 }
}

class Motorcycle extends Vehicle {
 @Override
 public void startEngine() {
     System.out.println("Motorcycle engine is starting with a kick...");
 }

 @Override
 public void stopEngine() {
     System.out.println("Motorcycle engine is stopping...");
 }
}


    public class Programm {
     public static void main(String[] args) {
             Vehicle myCar = new Car();
     myCar.startEngine();
     myCar.stopEngine();

     System.out.println();

     Vehicle myMotorcycle = new Motorcycle();
     myMotorcycle.startEngine();
     myMotorcycle.stopEngine();
}
}
```



```java
package in.cdac;

class Vehicle {
 public void startEngine() {
     System.out.println("Vehicle engine is starting...");
 }

 public void stopEngine() {
     System.out.println("Vehicle engine is stopping...");
 }
}

class Car extends Vehicle {
 @Override
 public void startEngine() {
     System.out.println("Car engine is starting with a key...");
 }

 @Override
 public void stopEngine() {
     System.out.println("Car engine is stopping...");
 }
}

class Motorcycle extends Vehicle {
 @Override
 public void startEngine() {
     System.out.println("Motorcycle engine is starting with a kick...");
 }

 @Override
 public void stopEngine() {
     System.out.println("Motorcycle engine is stopping...");
 }
}


    public class Programm {
```

```
<terminated> Programm [Java Application] C:\Users\Shreya\.p2\pool\plugins\org.ecli
Car engine is starting with a key...
Car engine is stopping...

Motorcycle engine is starting with a kick...
Motorcycle engine is stopping...
```