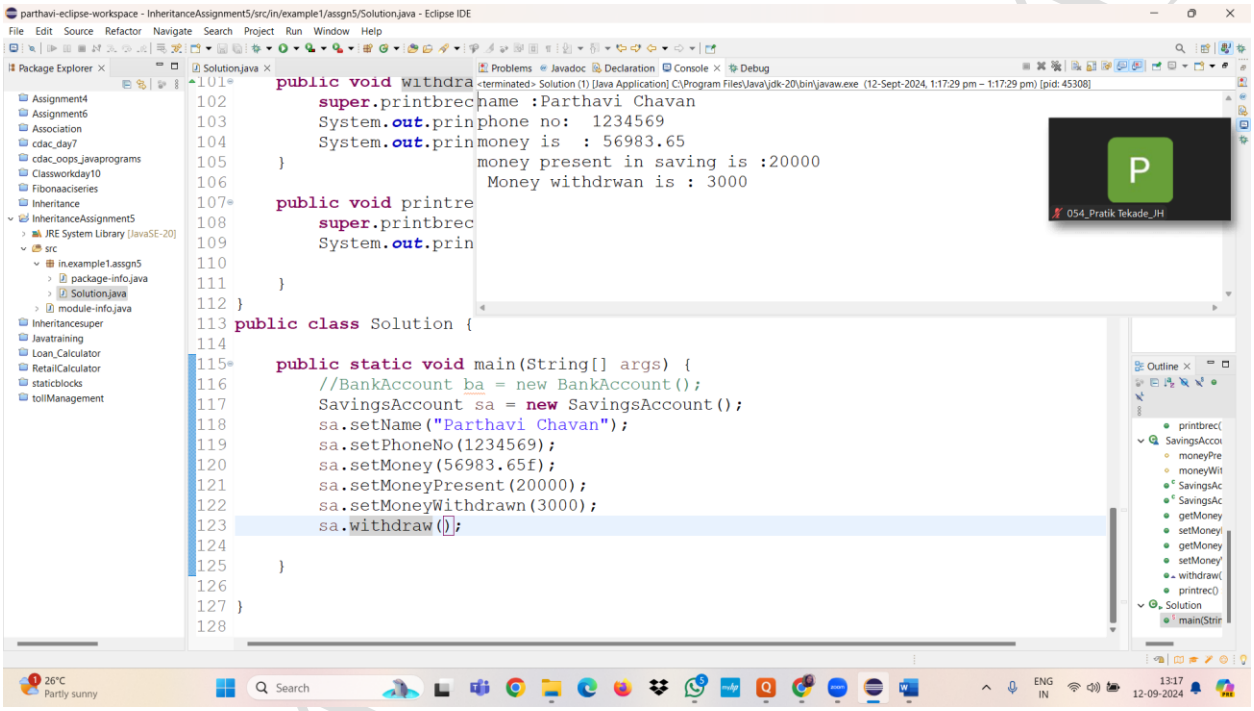
CDAC Mumbai PG-DAC August 24

Assignment No- 5

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.

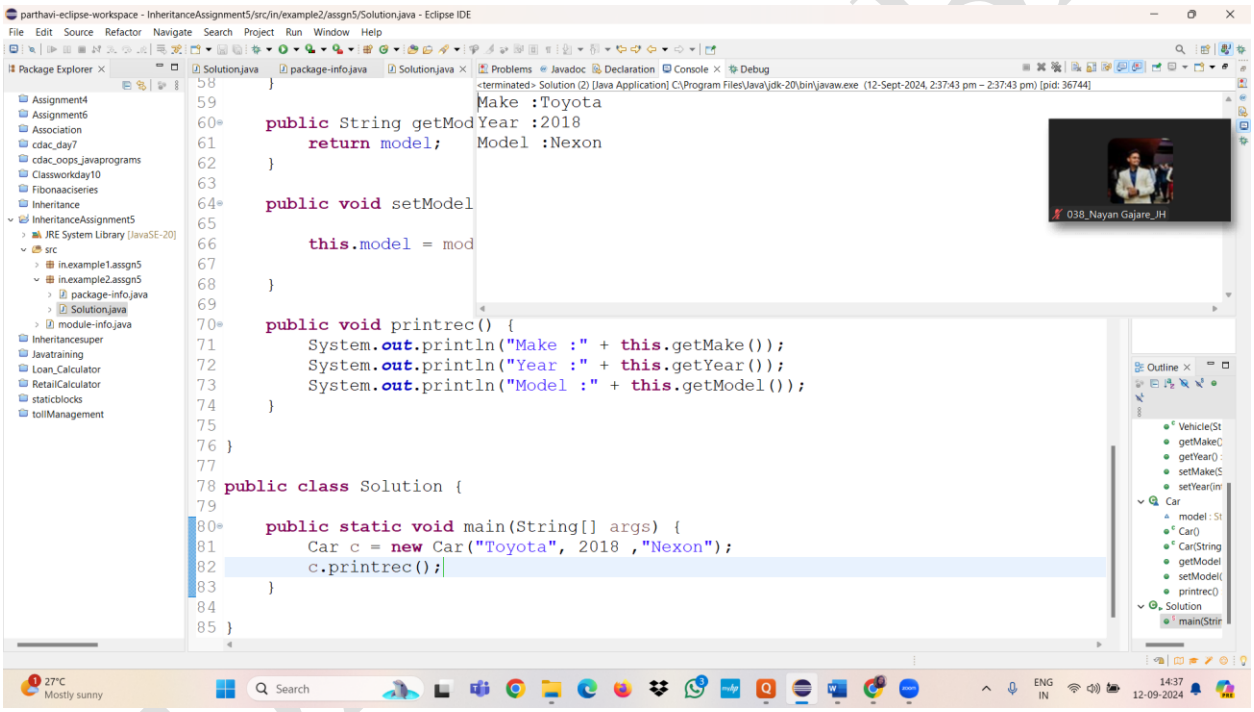


Create a base class Vehicle with attributes like make and year. Provide a constructor in Vehicle to initialize these attributes. Derive a class Car that has an additional attribute model and write a constructor that initializes make, year, and model. Write a program to create a Car object and display its details.

```
package in.example2.assgn5;
class Vehicle{

        String make;
        int year;

        public Vehicle() {
          this(" " , 0);
        }
```

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

          this.make = make ;
          this.year = year;
        }

        public String getMake() {
          return make;
        }

        public int getYear() {
          return year;
        }

        public void setMake(String make) {
          this.make = make;
        }

        public void setYear(int year) {
          this.year = year;
        }



}

class Car extends Vehicle{

        String model;

        public Car() {

        }

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

        }

        public String getModel() {
          return model;
        }

        public void setModel(String model) {

          this.model = model;

        }

        public void printrec() {
```

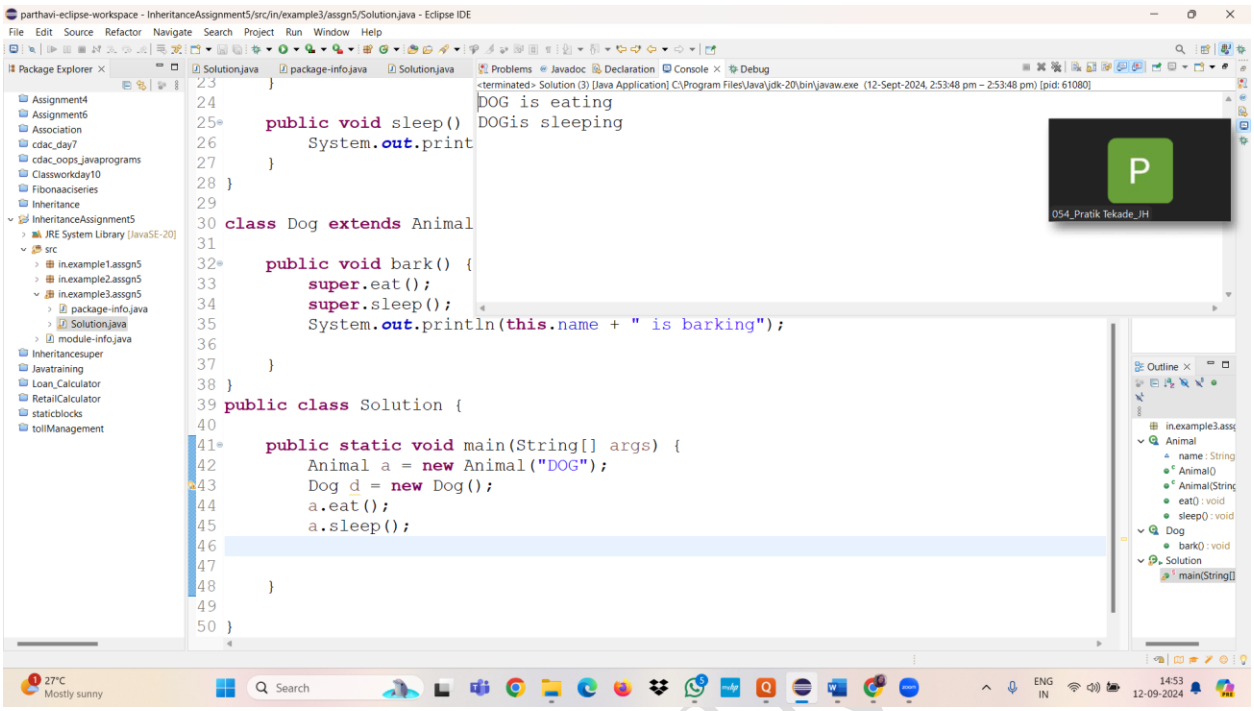
```
        System.out.println("Make :" + this.getMake());
        System.out.println("Year :" + this.getYear());
        System.out.println("Model :" + this.getModel());
    }

}

public class Solution {

    public static void main(String[] args) {
        Car c = new Car("Toyota", 2018 ,"Nexon");
        c.printrec();
    }

}
```



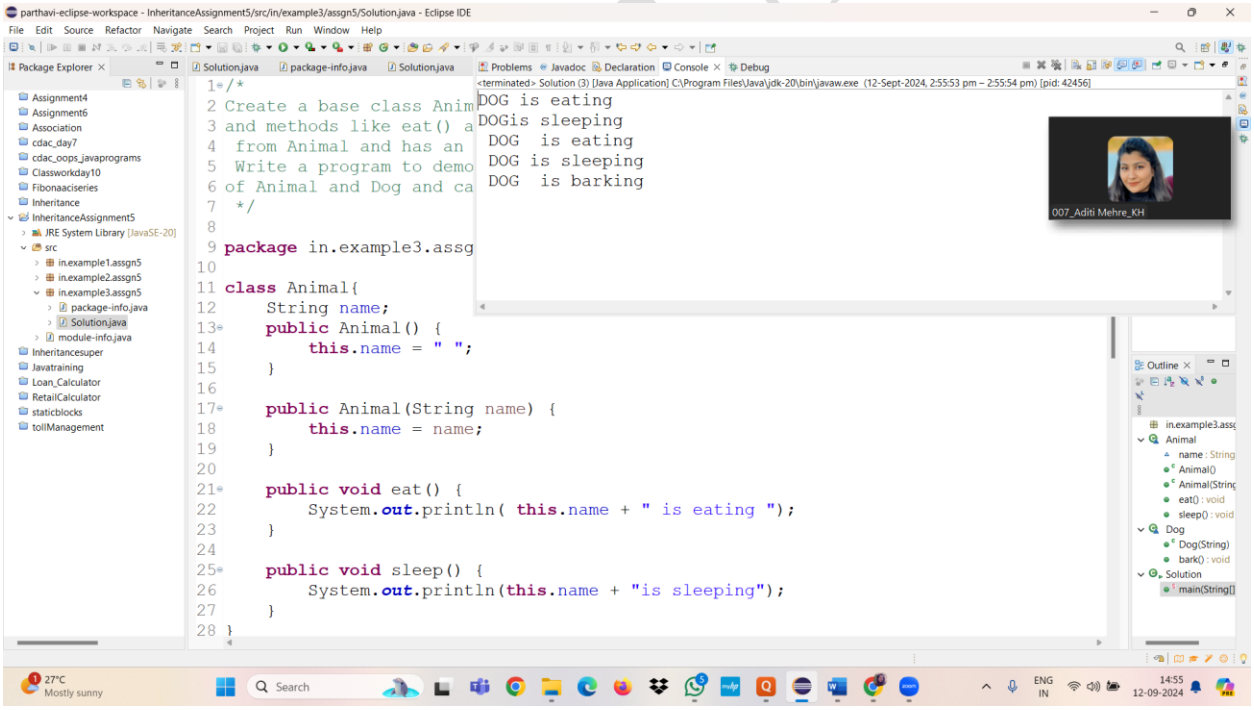
Create a base class Animal with attributes like name, and methods like eat() and sleep(). Create a subclass Dog that inherits from Animal and has an additional method bark(). Write a program to demonstrate the use of inheritance by creating objects of Animal and Dog and calling their methods.

```
package in.example3.assgn5;

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

        public Animal(String name) {
```

```
            this.name = name;
        }

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

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

class Dog extends Animal{

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

        public void bark() {
            super.eat();
            super.sleep();
            System.out.println(this.name + " is barking");

        }
}
public class Solution {

        public static void main(String[] args) {
            Animal a = new Animal("DOG");
            Dog d = new Dog(" DOG ");
            a.eat();
            a.sleep();
            d.bark();

        }

}
```




```
    }

    public void sleep() {
        System.out.print
    }
}

class Dog extends Animal

    public void bark() {
        super.eat();
        super.sleep();
        System.out.println(this.name + " is barking");

    }
}
public class Solution {

    public static void main(String[] args) {
        Animal a = new Animal("DOG");
        Dog d = new Dog();
        a.eat();
        a.sleep();


    }

}
```

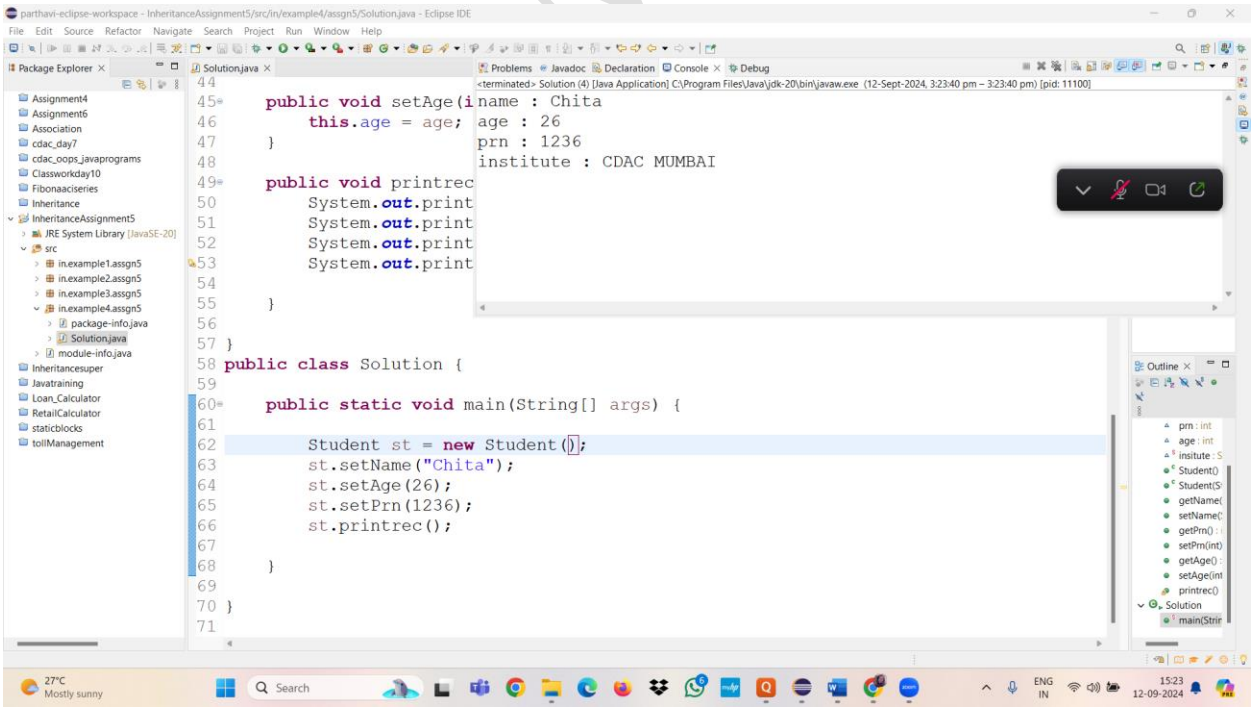



```
/*
Create a base class Anim
and methods like eat() a
 from Animal and has an
 Write a program to demo
of Animal and Dog and ca
 */

package in.example3.assg

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

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

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

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

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

```
/*
Build a class Student which contains details about the
Student and compile and run its
instance.
 */
package in.example4.assgn5;

class Student{
	String name;
	int prn;
	int age;
	static String insitute = "CDAC MUMBAI";

	public Student() {

	}

	public Student(String name, int prn, int age , String institute) {
		this.name = name;
		this.prn = prn;
		this.age = age;

	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public int getPrn() {
		return prn;
	}

	public void setPrn(int prn) {
		this.prn = prn;
	}

	public int getAge() {
		return age;
	}

	public void setAge(int age) {
```

```
            this.age = age;
        }

        public void printrec() {
            System.out.println("name : " + this.getName());
            System.out.println("age : " + this.getAge());
            System.out.println("prn : " + this.getPrn());
            System.out.println("institute : " + this.insitute);

        }

}
public class Solution {

        public static void main(String[] args) {

            Student st = new Student();
            st.setName("Chita");
            st.setAge(26);
            st.setPrn(1236);
            st.printrec();

        }

}
```



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.

```
/*
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.example5.assgn5;

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

	public void stopEngine() {
		System.out.println(" engine has stopped ");
	}
}

class Car extends Vehicle{

	@Override
	public void startEngine() {
		System.out.println(" engine is starting in car");
	}

	@Override
	public void stopEngine() {
		System.out.println(" engine has stopped in car ");
	}
}


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

	@Override
	public void stopEngine() {
		System.out.println(" engine has stopped in motorcycle ");
	}
```

```
}
public class Solution {

	public static void main(String[] args) {

		Vehicle vh = new Car();
		vh.startEngine();
		vh.stopEngine();
		vh = new Motorcycle();
		vh.startEngine();
		vh.stopEngine();

	}

}
```