CDAC
Mumbai PG-DAC August 24

Assignment No- 5

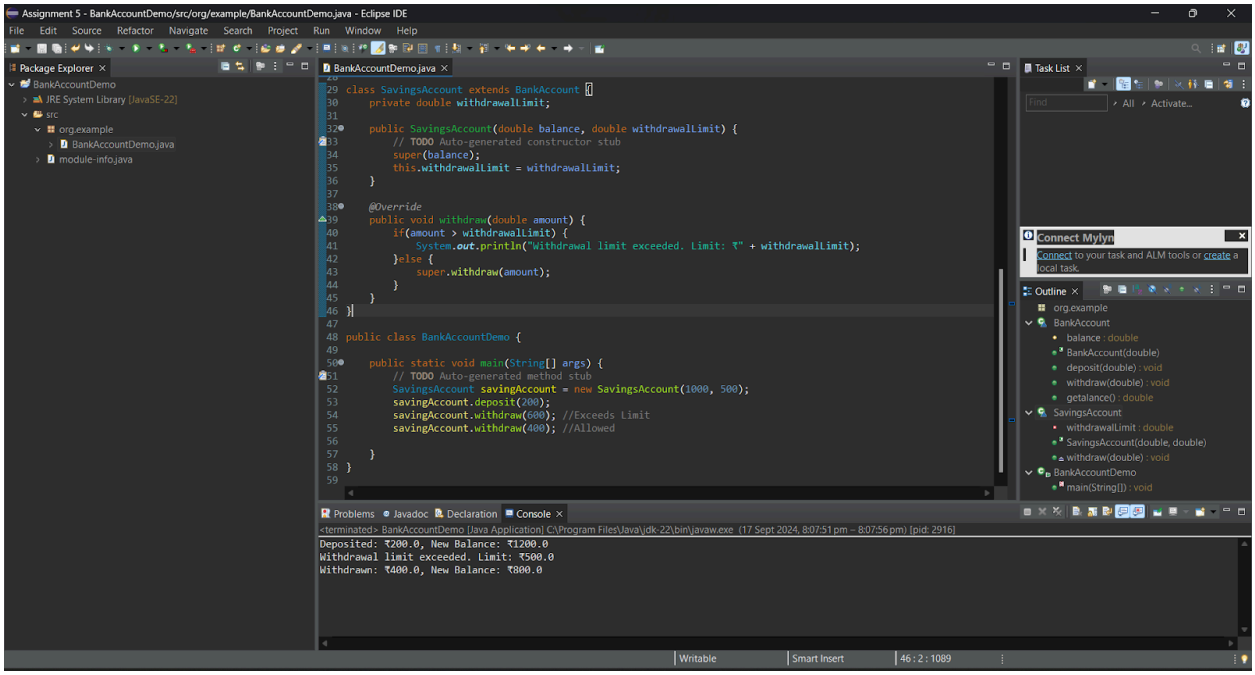
Name: Qureshi Mohammed Muqarrab
PRN No: 240840520036
CDAC Juhu

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.example;

class BankAccount {
	protected double balance;

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

	public void deposit(double amount) {
		balance += amount;
		System.out.println("Deposited: ₹" + amount + ", New Balance: ₹" + balance);
	}

	public void withdraw(double amount) {
		if(balance >= amount) {
			balance -= amount;
			System.out.println("Withdrawn: ₹" + amount + ", New Balance: ₹" + balance);
		}else {
			System.out.println("Insufficient funds. Withdrawal failed");
		}
	}

	public double getalance() {
		return balance;
	}
}
class SavingsAccount extends BankAccount {
	private double withdrawalLimit;
```

```java
	public SavingsAccount(double balance, double withdrawalLimit) {
		// TODO Auto-generated constructor stub
		super(balance);
		this.withdrawalLimit = withdrawalLimit;
	}

	@Override
	public void withdraw(double amount) {
		if(amount > withdrawalLimit) {
			System.out.println("Withdrawal limit exceeded. Limit: ₹" + withdrawalLimit);
		}else {
			super.withdraw(amount);
		}
	}
}
public class BankAccountDemo {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		SavingsAccount savingAccount = new SavingsAccount(1000, 500);
		savingAccount.deposit(200);
		savingAccount.withdraw(600); //Exceeds Limit
		savingAccount.withdraw(400); //Allowed
	}
}
```

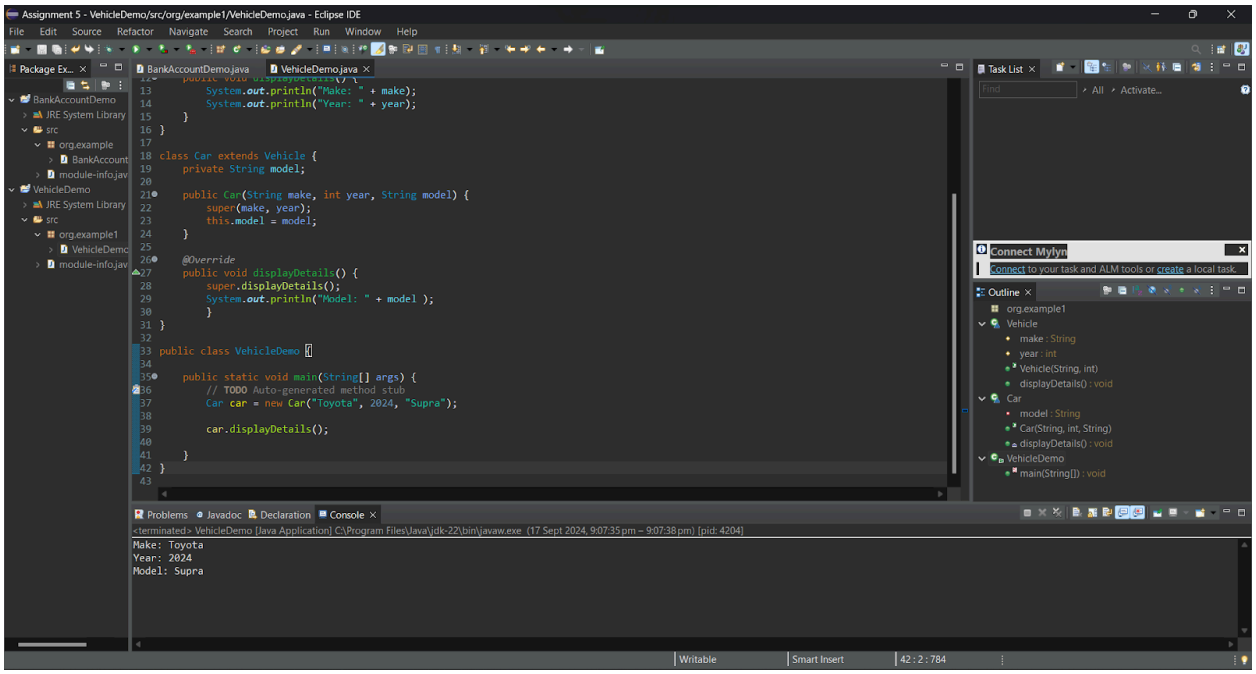
Output:



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.

```java
package org.example1;

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 VehicleDemo {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Car car = new Car("Toyota", 2024, "Supra");

		car.displayDetails();
	}
}
```

Output:

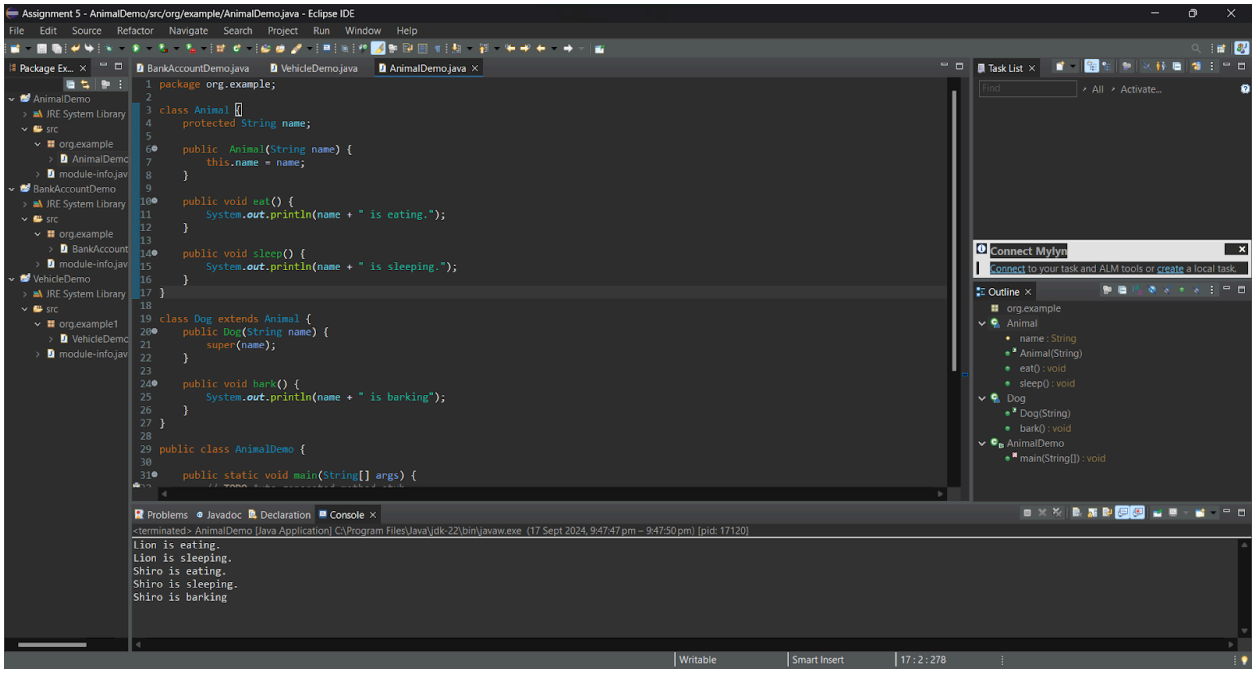

Assignment 5 - VehicleDemo/src/org/example1/VehicleDemo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Ex...
BankAccountDemo.java
VehicleDemo.java
Task List
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
Problems Javadoc Declaration Console
<terminated> VehicleDemo [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (17 Sept 2024, 9:07:35 pm – 9:07:38 pm) [pid: 4204]
Make: Toyota
Year: 2024
Model: Supra
Writable
Smart Insert
42 : 2 : 784

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.

```java
package org.example;

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 AnimalDemo {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Animal animal = new Animal("Lion");
		animal.eat();
		animal.sleep();

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

```
}
```

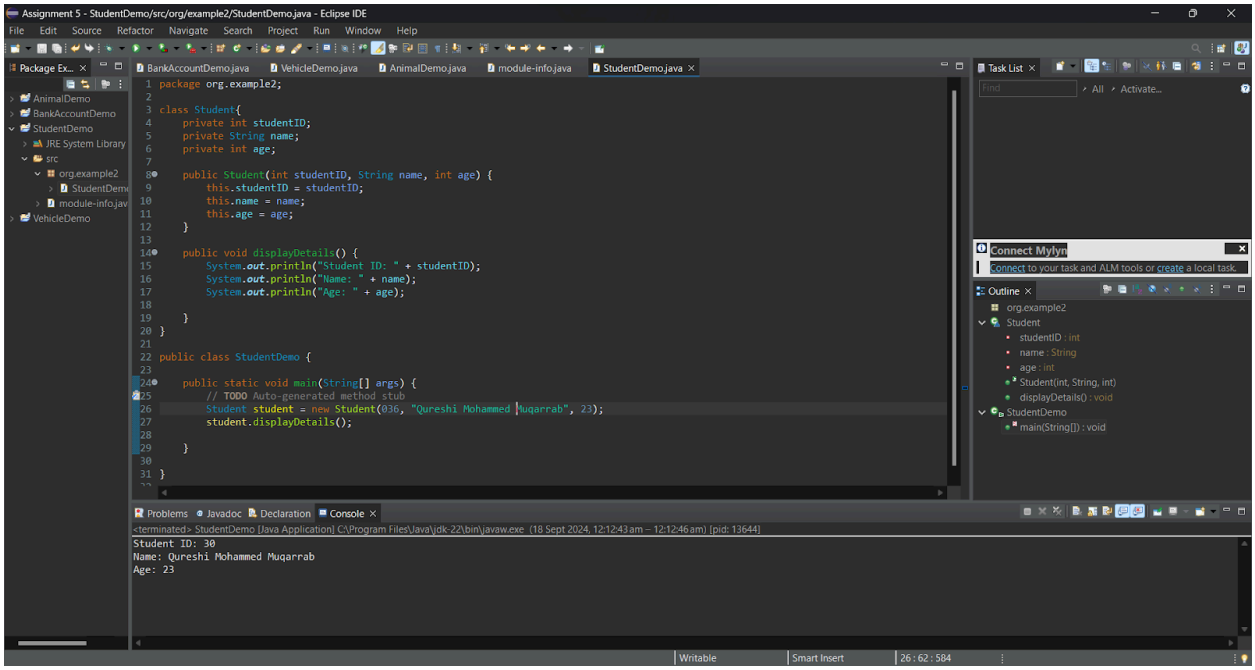
Output:


Assignment 5 - AnimalDemo/src/org/example/AnimalDemo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
BankAccountDemo.java
VehicleDemo.java
AnimalDemo.java
package org.example;
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 AnimalDemo {
public static void main(String[] args) {
Problems Javadoc Declaration Console
<terminated> AnimalDemo [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (17 Sept 2024, 9:47:47 pm – 9:47:50 pm) [pid: 17120]
Lion is eating.
Lion is sleeping.
Shiro is eating.
Shiro is sleeping.
Shiro is barking
Writable
Smart Insert
17 : 2 : 278

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

```
package org.example2;
class Student{
	private int studentID;
	private String name;
	private int age;

	public Student(int studentID, String name, int age) {
		this.studentID = studentID;
		this.name = name;
		this.age = age;
	}

	public void displayDetails() {
		System.out.println("Student ID: " + studentID);
		System.out.println("Name: " + name);
		System.out.println("Age: " + age);
	}
}
public class StudentDemo {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Student student = new Student(036, "Qureshi Mohammed Muqarrab", 23);
		student.displayDetails();
	}
}
```

Output:



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 org.example3;
class Vehicle {
  public void startEngine() {
     System.out.println("Vehicle engine is starting.");       //start vehicle
  }
  public void stopEngine() {
     System.out.println("Vehicle engine is stopping.");      //stop vehicle
  }
}
class Car extends Vehicle {
  @Override
  public void startEngine() {                  //classs car extends vehicle (Car is a vehicle)
     System.out.println("Car engine is starting with a key.");
  }
  @Override
  public void stopEngine() {
     System.out.println("Car engine is stopping.");                    //override methods for car
  }
}
class Motorcycle extends Vehicle {
  @Override
  public void startEngine() {
     System.out.println("Motorcycle engine is starting with a button.");
  }
  @Override
```

```java
  public void stopEngine() {        //override-method          (Motorcycle is a vehicle)
      System.out.println("Motorcycle engine is stopping.");
  }
}
public class VehicleDemo2 {       //main class and method
  public static void main(String[] args) {
      Vehicle car = new Car();
      car.startEngine();
      car.stopEngine();
      Vehicle motorcycle = new Motorcycle();
      motorcycle.startEngine();
      motorcycle.stopEngine();
  }
}
```

Output: