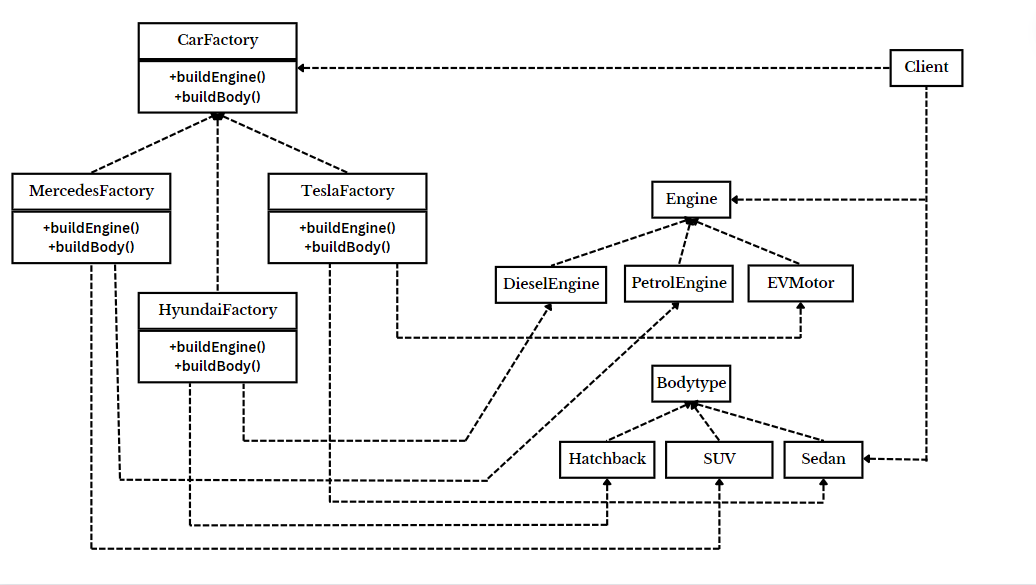
**Abstract Design Pattern**

**Assignment - 2**

Name - Heet Dobariya Roll No. - 22BCP177 Group - G5

* **Program :** Create a Abstract Design Pattern for suitable example.
* **UML Diagram :**



* **Code :**

package EnginewithBodytype;

interface Engine

{

void displayEngine();

}

package EnginewithBodytype;

// Concrete Products of Engine: PetrolEngine, DieselEngine, EVMotor

class PetrolEngine implements Engine

{

@Override

public void displayEngine()

{

System.out.println("Powered by Petrol, which gives mind blowing thrust!");

}

}

package EnginewithBodytype;

class DieselEngine implements Engine

{

@Override

public void displayEngine()

{

System.out.println("Powered by Diesel, which is bit more polluting!");

}

}

package EnginewithBodytype;

class EVMotor implements Engine

{

@Override

public void displayEngine()

{

System.out.println("Powered by Electricity, Huhh kids!");

}

}

package EnginewithBodytype;

interface BodyType

{

void displayBody();

}

package EnginewithBodytype;

class SUV implements BodyType

{

@Override

public void displayBody()

{

System.out.println("SUV: Having a body roll!");

}

}

package EnginewithBodytype;

class Sedan implements BodyType

{

@Override

public void displayBody()

{

System.out.println("Sedan: Most stable on road!");

}

}

package EnginewithBodytype;

class Hatchback implements BodyType

{

@Override

public void displayBody()

{

System.out.println("Hatchback: Known for compactness!");

}

}

package EnginewithBodytype;

interface CarFactory

{

Engine buildEngine();

BodyType buildBody();

}

package EnginewithBodytype;

class MercedesFactory implements CarFactory

{

@Override

public Engine buildEngine()

{

return new PetrolEngine();

}

@Override

public BodyType buildBody()

{

return new SUV();

}

}

package EnginewithBodytype;

class HyundaiFactory implements CarFactory

{

@Override

public Engine buildEngine()

{

return new DieselEngine();

}

@Override

public BodyType buildBody()

{

return new Hatchback();

}

}

package EnginewithBodytype;

class TeslaFactory implements CarFactory

{

@Override

public Engine buildEngine()

{

return new EVMotor();

}

@Override

public BodyType buildBody()

{

return new Sedan();

}

}

package EnginewithBodytype;

public class Client

{

public static void main(String[] args)

{

System.out.println("For Petrol SUVs:");

CarFactory mercedesFactory = new MercedesFactory();

Engine petrolEngine = mercedesFactory.buildEngine();

BodyType suv = mercedesFactory.buildBody();

petrolEngine.displayEngine();

suv.displayBody();

System.out.println("For Diesel Hatchbacks:");

CarFactory suzukiFactory = new HyundaiFactory();

Engine dieselEngine = suzukiFactory.buildEngine();

BodyType hatchback = suzukiFactory.buildBody();

dieselEngine.displayEngine();

hatchback.displayBody();

System.out.println("For EV Sedans:");

CarFactory teslaFactory = new TeslaFactory();

Engine evMotor = teslaFactory.buildEngine();

BodyType sedan = teslaFactory.buildBody();

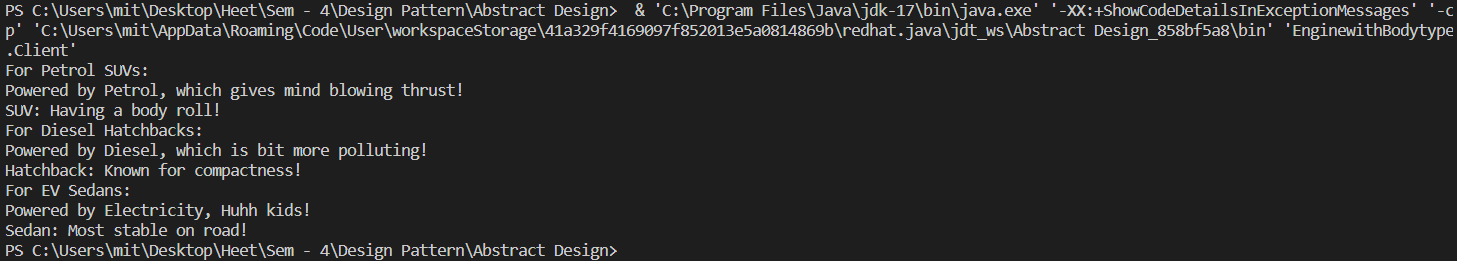
evMotor.displayEngine();

sedan.displayBody();

}

}

* **Output :**

****