DAY 5 PRACTICE EXERCISE

1. **package** usha;

**public** **class** BillingService {

**private** **static** BillingService *instance*;

**private** BillingService() {

}

**public** **static** BillingService getinstance() {

**if**(*instance*==**null**) {

*instance*=**new** BillingService();

}

**return** *instance*;

}

**public** **void** processPayment(String paymentDetails) {

System.***out***.println("Processing payment:"+paymentDetails);

}

**public** **void** generateinvoice(String orderDetails) {

System.***out***.println("generating invoice for order:"+orderDetails);

}

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

// **TODO** Auto-generated method stub

}

}

**public** **class** Main1 {

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

// **TODO** Auto-generated method stub

BillingService billingService=BillingService.*getinstance*();

billingService.processPayment("Payment details:$100");

billingService.generateinvoice("Order details:Product A,Quantity:2");

}

}

2. 2.1**package** usha;

**public** **class** Car **implements** Vehicle {

**public** **void** start() {

System.***out***.println("Car started");

}

**public** **void** accelerate() {

System.***out***.println("car is accelerate");

}

**public** **void** brake() {

System.***out***.println("car is brake");

}

}

2.2 **public** **class** Motorcycle **implements** Vehicle {

**public** **void** start() {

System.***out***.println("Motorcycle started");

}

**public** **void** acelerate() {

System.***out***.println("Motorcycle is acclerate");

}

**public** **void** brake() {

System.***out***.println("Motorcycle is brake");

}

}

2.3 **public** **class** Truck **implements** Vehicle {

**public** **void** start() {

System.***out***.println("Truck start");

}

**public** **void** accelerate() {

System.***out***.println("Truck is accelerate");

}

**public** **void** brake() {

System.***out***.println("Truck is brake");

}

}

2.4 **public** **class** VehicleFactory {

**public** Vehicle CreateVehicle(String type) {

**if**(type.equalsIgnoreCase("car")) {

**return** **new** Car();

}

**else** **if** (type.equalsIgnoreCase("motorcycle")) {

**return** **new** Motorcycle();

}

**else** **if**(type.equalsIgnoreCase("truck")) {

**return** **new** Truck();

}

**return** **null**;

}

}

2.5 **public** **class** Main2 {

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

// **TODO** Auto-generated method stub

VehicleFactory factory =**new** VehicleFactory();

Vehicle car = factory.CreateVehicle("car");

**if**(car!=**null**) {

car.start();

car.accelerate();

car.brake();

}

Vehicle motorcycle = factory.CreateVehicle("motorcycle");

**if**(motorcycle!=**null**) {

motorcycle.start();

motorcycle.accelerate();

motorcycle.brake();

}

Vehicle Truck= factory.CreateVehicle("Truck");

**if**(Truck!=**null**) {

Truck.start();

Truck.accelerate();

Truck.brake();

}

}

}

3. **package** usha;

**public** **class** Shape {

**public** **interface** shape{

**void** draw();

}

**public** **class** Circle **implements** shape{

@Override

**public** **void** draw () {

System.***out***.println("Inside Circle::draw() method.");

}

}

//Rectangle.java

**public** **class** Rectangle **implements** shape{

@Override

**public** **void** draw() {

System.***out***.println("Inside Rectangle::draw() method.");

}

}

// Square.java

**public** **class** Square **implements** shape{

@Override

**public** **void** draw() {

System.***out***.println("Inside Square::draw() method.");

}

}

//AbstractFactory.java

**public** **abstract** **class** AbstractFactory{

**abstract** shape getshape(String shapeType);

}

//shapefactory.java

**public** **class** shapeFactory **extends** AbstractFactory{

@Override

**public** shape getshape(String shapeType) {

**if**(shapeType==**null**) {

**return** **null**;

}

**if**(shapeType.equalsIgnoreCase("Circle")) {

**return** **new** Circle();

}

**else** **if**(shapeType.equalsIgnoreCase("Rectangle")) {

**return** **new** Rectangle();

}

**else** **if**(shapeType.equalsIgnoreCase("Square")) {

**return** **new** Square();

}

**return** **null**;

}

//AbstractFactoryPattern.java

**public** **class** AbstractFactoryPattern{

**private** **static** Object *FactoryProducer*;

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

AbstractFactory shapeFactory=*FactoryProducer*.getFactory(**false**);

shape shape1 = shapeFactory.getshape("Circle");

shape1.draw();

shape shape2=shapeFactory.getshape("Rectanle");

shape2.draw();

shape shape3 = shapeFactory.getshape("Square");

shape3.draw();

}

}

//FactoryProducer.java

**public** **class** FactoryProducer{

**public** **static** AbstractFactory getFactor(**boolean** rounded) {

**return** **new** shapeFactory();

}

}

}

4 **package** usha;

**import** java.util.Date;

**public** **final** **class** Employee {

**private** **final** String firstName;

**private** **final** String lastName;

**private** **final** Date dateOfBirth;

**private** **final** **int** employeeId;

**private** **final** Date joiningDate;

**private** **final** **double** salary;

**public** Employee(String firstName, String lastName, Date dateOfBirth, **int** employeeId, Date joiningDate, **double** salary) {

**this**.firstName = firstName;

**this**.lastName = lastName;

**this**.dateOfBirth = **new** Date(dateOfBirth.getTime());

**this**.employeeId = employeeId;

**this**.joiningDate = **new** Date(joiningDate.getTime());

**this**.salary = salary;

   }

**public** String getFirstName() {

**return** firstName;

   }

**public** String getLastName() {

**return** lastName;

   }

**public** Date getDateOfBirth() {

**return** **new** Date(dateOfBirth.getTime());

   }

**public** **int** getEmployeeId() {

**return** employeeId;

   }

**public** Date getJoiningDate() {

**return** **new** Date(joiningDate.getTime());

   }

**public** **double** getSalary() {

**return** salary;

   }

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

Date dob = **new** Date(1990, 1, 1);

Date joiningDate = **new** Date(2020, 1, 1);

Employee employee = **new** Employee("John", "Doe", dob, 12345, joiningDate, 50000.0);

System.out.println("Employee Details:");

System.out.println("First Name: " + employee.getFirstName());

System.out.println("Last Name: " + employee.getLastName());

System.out.println("Date of Birth: " + employee.getDateOfBirth());

System.out.println("Employee ID: " + employee.getEmployeeId());

System.out.println("Joining Date: " + employee.getJoiningDate());

System.out.println("Salary: " + employee.getSalary());

}

}