**Name**: Ravi Jamanbhai Makwana

**Roll No**: 21BCP418

Div-6, G-12

**Assignment-1 (Factory Pattern)**

**Aim:**

To implement Factory Design Pattern using Accommodation management system.

**Code:**

room.java

public interface room{

    void allot();

}

single room.java

public class single\_room implements room{

    @Override

    public void allot(){

        System.out.println("Room for single person alloted");

    }

}

double\_room.java

public class double\_room implements room{

    @Override

    public void allot(){

        System.out.println("Room for 2 persons alloted");

    }

}

quadruple\_room.java

public class quadruple\_room implements room{

    @Override

    public void allot(){

        System.out.println("Room for 4 persons alloted");

    }

}

RoomFactory.java

public class RoomFactory {

    public room getRoom(String roomType){

        if (roomType == null){

            return null;

        }

        if (roomType.equalsIgnoreCase("single")){

            return new single\_room();

        }

        else if (roomType.equalsIgnoreCase("double")){

            return new double\_room();

        }

        else if (roomType.equalsIgnoreCase("quadruple")){

            return new quadruple\_room();

        }

        else{

            return null;

        }

    }

}

Test.java

public class Test{

    public static void main(String args[]) throws NullPointerException{

        RoomFactory rf = new RoomFactory();

        try{

            room r1 = rf.getRoom("SINGLE");

            r1.allot();

            room r2 = rf.getRoom("QUADRUPLE");

            r2.allot();

            room r3 = rf.getRoom("DOUBLE");

            r3.allot();

            room r4 = rf.getRoom("Ravi");

            r4.allot();

        }

        catch (NullPointerException ex){

            System.out.println("Room not available!");

        }

    }

}

**Output:**

