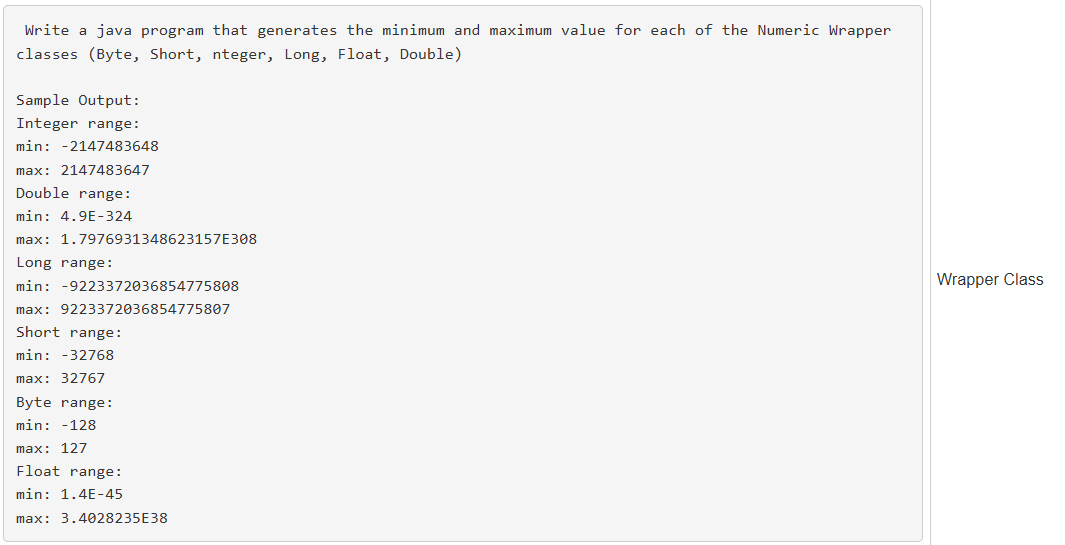
**BYAGARI PAVAN PBL ID: J\_251890123**

Wrapper Class

# Code:-

public class WrapperRange {

public static void main(String[] args) {

System.out.println("Integer range:");

System.out.println("min: " + Integer.MIN\_VALUE);

System.out.println("max: " + Integer.MAX\_VALUE);

System.out.println("Double range:");

System.out.println("min: " + Double.MIN\_VALUE);

System.out.println("max: " + Double.MAX\_VALUE);

System.out.println("Long range:");

System.out.println("min: " + Long.MIN\_VALUE);

System.out.println("max: " + Long.MAX\_VALUE);

System.out.println("Short range:");

System.out.println("min: " + Short.MIN\_VALUE);

System.out.println("max: " + Short.MAX\_VALUE);

System.out.println("Byte range:");

System.out.println("min: " + Byte.MIN\_VALUE);

System.out.println("max: " + Byte.MAX\_VALUE);

System.out.println("Float range:");

System.out.println("min: " + Float.MIN\_VALUE);

System.out.println("max: " + Float.MAX\_VALUE);

}

}

# Output:-

Integer range:

min: -2147483648

max: 2147483647

Double range:

min: 4.9E-324

max: 1.7976931348623157E308

Long range:

min: -9223372036854775808

max: 9223372036854775807

Short range:

min: -32768

max: 32767

Byte range:

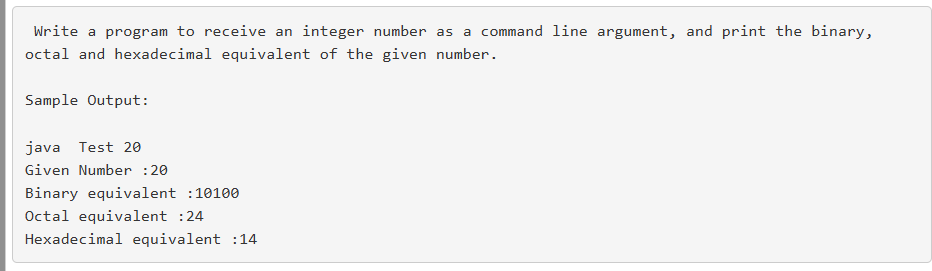
min: -128

max: 127

Float range:

min: 1.4E-45

max: 3.4028235E38



# Code:-

public class Test {

public static void main(String[] args) {

Integer num = Integer.parseInt(args[0]);

System.out.println("Given Number: " + num);

System.out.println("Binary equivalent: " + Integer.toBinaryString(num));

System.out.println("Octal equivalent : " + Integer.toOctalString(num));

System.out.println("Hexadecimal equivalent : " + Integer.toHexString(num));

}

}

# Execution & Output:-

Command:-

java Test 20

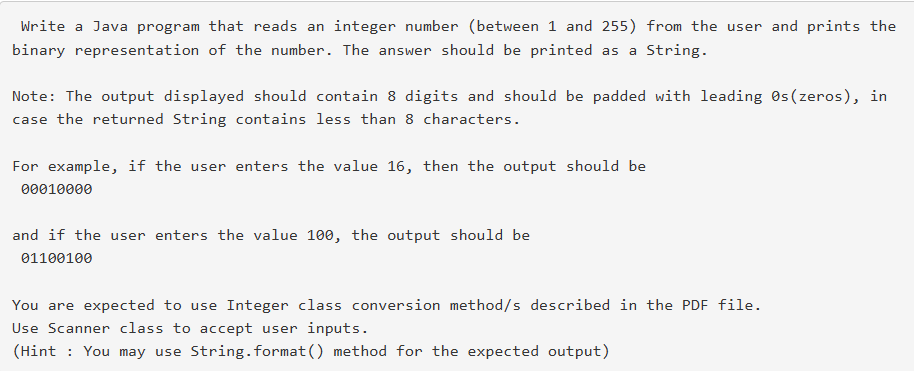
Output:-

Given Number: 20

Binary equivalent: 10100

Octal equivalent : 24

Hexadecimal equivalent : 14



# Code:-

import java.util.Scanner;

public class BinaryFormat {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter an integer (1-255): ");

int num = sc.nextInt();

String binary = Integer.toBinaryString(num);

String padded = String.format("%8s", binary).replace(' ', '0');

System.out.println("Binary representation: " + padded);

}

}

Run & Output:-

Input:-

Enter an integer (1-255): 16

Output:-

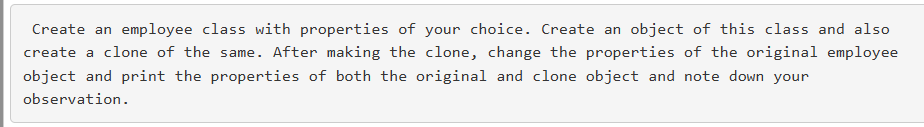
Binary representation: 00010000

Input:-

Enter an integer (1-255): 100

**Output:-**

Binary representation: 01100100



# Code:-

class Employee implements Cloneable {

int id;

String name;

double salary;

Employee(int id, String name, double salary) {

this.id = id;

this.name = name;

this.salary = salary;

}

public Object clone() throws CloneNotSupportedException {

return super.clone();

}

public void display() {

System.out.println("ID: " + id + ", Name: " + name + ", Salary: " + salary);

}

}

public class Main {

public static void main(String[] args) {

try {

Employee e1 = new Employee(101, "Pavan", 50000);

Employee e2 = (Employee) e1.clone();

System.out.println("Before changing original object:");

System.out.print("Original: ");

e1.display();

System.out.print("Clone : ");

e2.display();

e1.id = 102;

e1.name = "Deepu";

e1.salary = 60000;

System.out.println("\nAfter changing original object:");

System.out.print("Original: ");

e1.display();

System.out.print("Clone : ");

e2.display();

} catch (CloneNotSupportedException e) {

e.printStackTrace();

}

}

}

# Output:-

Before changing original object:

Original: ID: 101, Name: Pavan, Salary: 50000.0

Clone : ID: 101, Name: Pavan, Salary: 50000.0

After changing original object:

Original: ID: 102, Name: Deepu, Salary: 60000.0

Clone : ID: 101, Name: Pavan, Salary: 50000.0