Object Oriented Programming
(ITRC2)

IT III Semester

Submitted by

HIMANSHU GAWALI

23I3025

Information Technology -A

Submitted to

Aditya Makwe

Assistant Professor, Department of Information Technology

Institute of Engineering and Technology

Devi Ahilya Vishwavidhyalaya, Indore (M.P.) India

(www.iet.dauniv.ac.in**)**

Session July-Dec, 2024

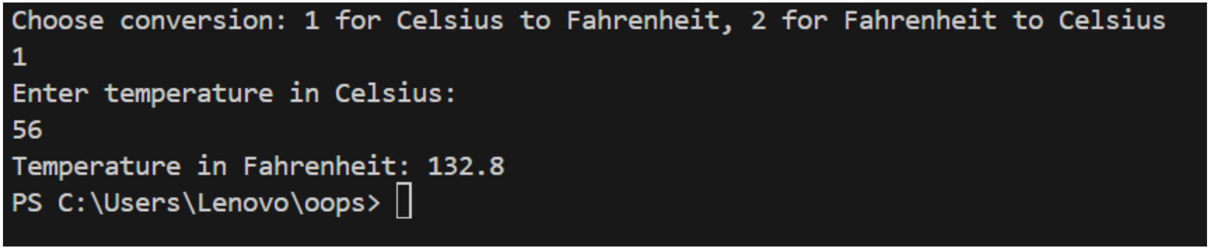
ASSIGNMENT-1

1. Write a Java program to change temperature from Celsius to Fahrenheit and vice versa.

Input-

```
import java.util.Scanner;

public class temperatureconverter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Choose conversion: 1 for Celsius to Fahrenheit, 2 for Fahrenheit to
Celsius");
        int choice = scanner.nextInt();

        if (choice == 1) {
            System.out.println("Enter temperature in Celsius:");
            double celsius = scanner.nextDouble();
            double fahrenheit = (celsius * 9/5) + 32;
            System.out.println("Temperature in Fahrenheit: " + fahrenheit);
        } else if (choice == 2) {
            System.out.println("Enter temperature in Fahrenheit:");
            double fahrenheit = scanner.nextDouble();
            double celsius = (fahrenheit - 32) * 5/9;
            System.out.println("Temperature in Celsius: " + celsius);
        } else {
            System.out.println("Invalid choice");
        }

        scanner.close();
    }
}
```

Output-



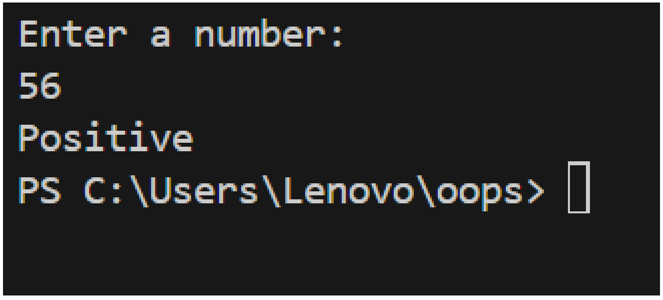
```
Choose conversion: 1 for Celsius to Fahrenheit, 2 for Fahrenheit to Celsius
1
Enter temperature in Celsius:
56
Temperature in Fahrenheit: 132.8
PS C:\Users\Lenovo\oops> 
```

2. Write a Java Program to check if a number is Positive or Negative.

Input-

```
import java.util.Scanner;

public class PositiveNegative {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a number:");
        int number = scanner.nextInt();

        if (number > 0) {
            System.out.println("Positive");
        } else if (number < 0) {
            System.out.println("Negative");
        } else {
            System.out.println("Zero");
        }

        scanner.close();
    }
}
```

Output-



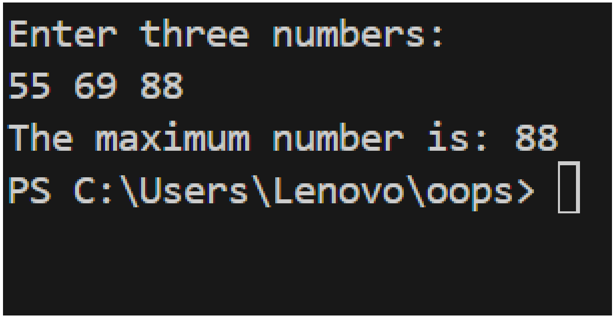
```
Enter a number:
56
Positive
PS C:\Users\Lenovo\oops>
```

3. Write a Java program to find maximum of three numbers.

Input-

```
import java.util.Scanner;

public class MaximumOfThree {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter three numbers:");

        int num1 = scanner.nextInt();
        int num2 = scanner.nextInt();
        int num3 = scanner.nextInt();

        int max = num1;

        if (num2 > max) {
            max = num2;
        }
        if (num3 > max) {
            max = num3;
        }

        System.out.println("The maximum number is: " + max);

        scanner.close();
    }
}
```

Output-



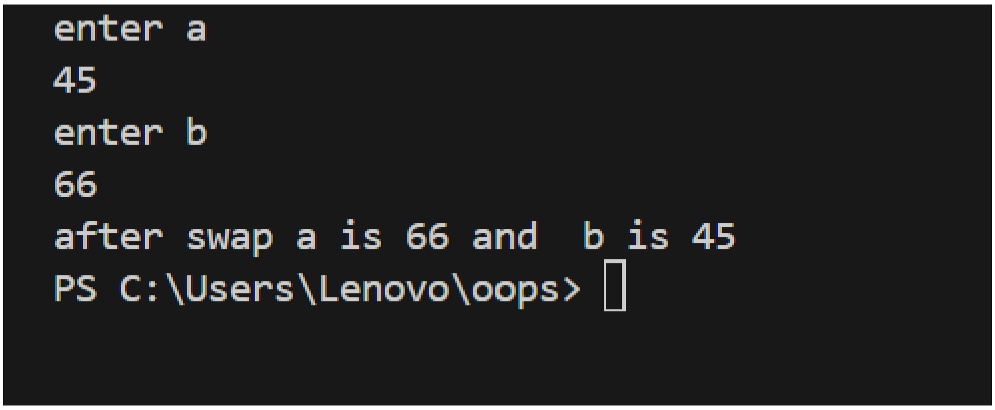
```
Enter three numbers:
55 69 88
The maximum number is: 88
PS C:\Users\Lenovo\oops>
```

4. Write a Java program to swap two numbers.

Input-

```
import java.util.Scanner;

public class swap{

    public static void main(String[] args) {
        int a = 2;
        int b = 3;

        int temp = a;
        a = b;
        b = temp;

        System.out.println("a = " + a);
        System.out.println("b = " + b);
    }
}
```

Output-



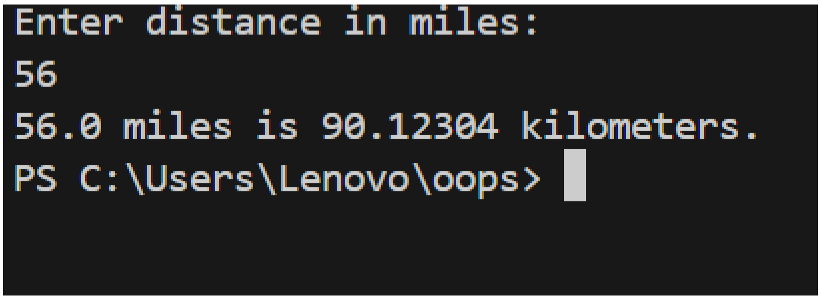
```
enter a
45
enter b
66
after swap a is 66 and  b is 45
PS C:\Users\Lenovo\oops> 
```

5. Write a Java program to convert miles to kilometers.

Input-

```
import java.util.Scanner;

public class converter{
    public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      System.out.println("Enter distance in miles:");
      double miles = scanner.nextDouble();

      double kilometers = miles * 1.60934;
      System.out.println(miles + " miles is " + kilometers + " kilometers.");


   }
}
```

Output_



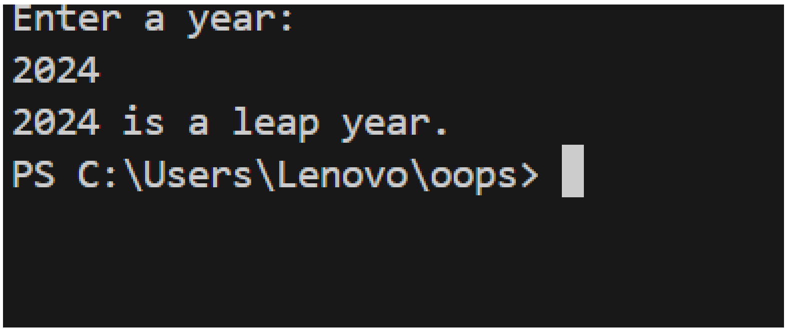
```
Enter distance in miles:
56
56.0 miles is 90.12304 kilometers.
PS C:\Users\Lenovo\oops>
```

6. Write a Java program to check whether a year is leap year or not.

Input-

```
import java.util.Scanner;

public class leapyear {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter a year:");
        int year = scanner.nextInt();

        if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) {
            System.out.println(year + " is a leap year.");
        } else {
            System.out.println(year + " is not a leap year.");
        }
    }
}
```

Output-



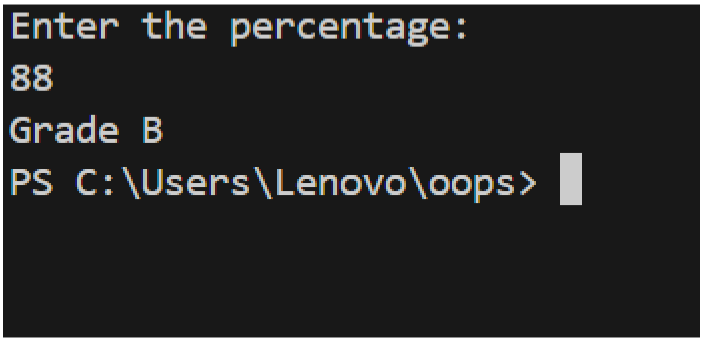
```
Enter a year:
2024
2024 is a leap year.
PS C:\Users\Lenovo\oops>
```

7. Write a Java program for following grading system.Note:
Percentage>=90% : Grade A
Percentage>=80% : Grade B
Percentage>=70% : Grade C
Percentage>=60% : Grade D
Percentage>=40% : Grade E
Percentage<40% : Grade F

Input-

```
import java.util.Scanner;
public class percent {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.println("Enter the percentage:");
    double percentage = scanner.nextDouble();

    if (percentage >= 90) {
      System.out.println("Grade A");
    } else if (percentage >= 80) {
      System.out.println("Grade B");
    } else if (percentage >= 70) {
      System.out.println("Grade C");
    } else if (percentage >= 60) {
      System.out.println("Grade D");
    } else if (percentage >= 40) {
      System.out.println("Grade E");
    } else {
      System.out.println("Grade F");
    }
  }
}
```

Output-

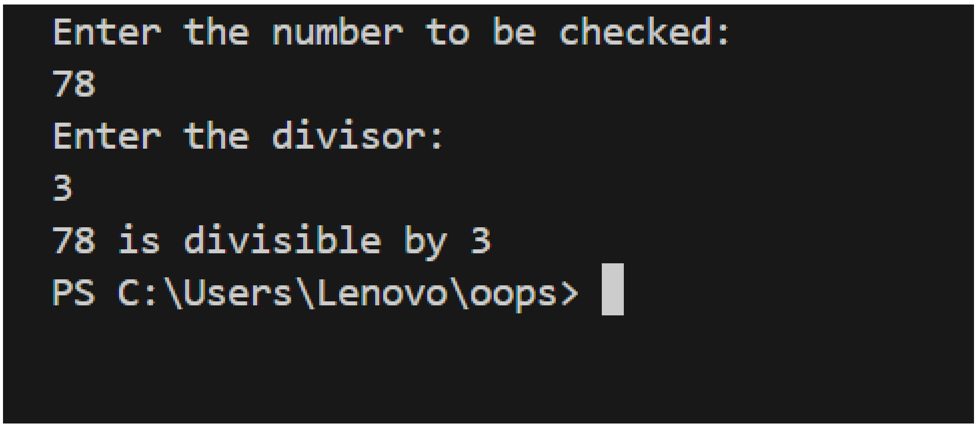


```
Enter the percentage:
88
Grade B
PS C:\Users\Lenovo\oops>
```

8. Write a Java program to check whether a number is divisible by a number given by user

Input-

```
import java.util.Scanner;

public class division {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the number to be checked:");
        int number = scanner.nextInt();

        System.out.println("Enter the divisor:");
        int divisor = scanner.nextInt();

        if (divisor != 0) {
            if (number % divisor == 0) {
                System.out.println(number + " is divisible by " + divisor);
            } else {
                System.out.println(number + " is not divisible by " + divisor);
            }
        } else {
            System.out.println("Divisor cannot be zero.");
        }
    }
}
```



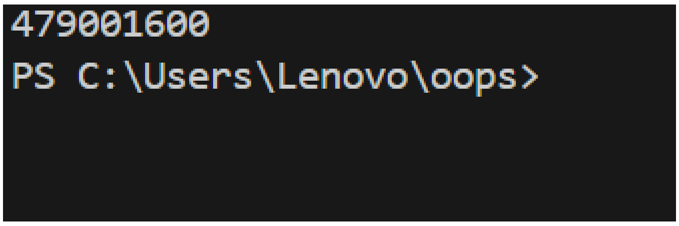
```
Enter the number to be checked:
78
Enter the divisor:
3
78 is divisible by 3
PS C:\Users\Lenovo\oops> 
```

9. Write a Java program to calculate factorial of 12.

Input-

```
import java.util.Scanner;

public class Q9 {
    public static void main(String[] args) {
        int number = 12;
        int factorial = 1;

        if (number == 0 || number == 1) {
            factorial = 1;
        } else {
            for (int i = 2; i <= number; i++) {
                factorial *= i;
            }
        }

        System.out.println(factorial);
    }
}
```

Output-



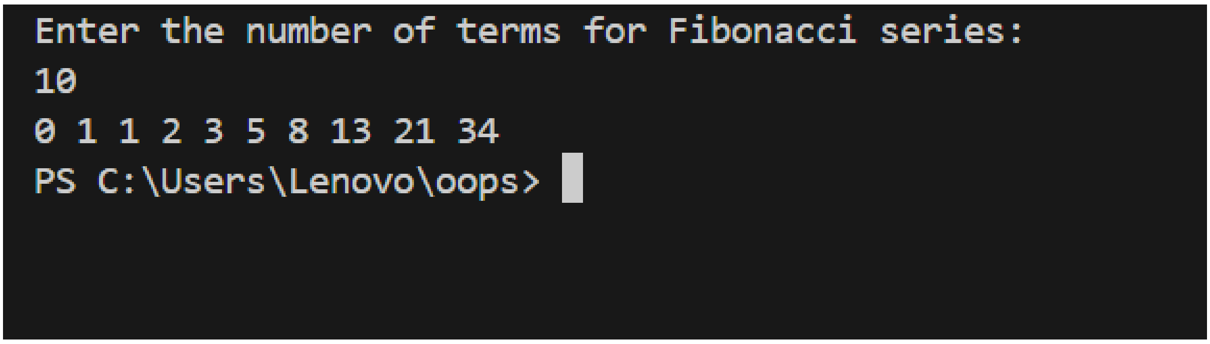
```
479001600
PS C:\Users\Lenovo\oops>
```

10. Write a Java program for Fibonacci series.

Input-

```
import java.util.Scanner;

public class FibonacciSeries {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of terms for Fibonacci series: ");
        int n = scanner.nextInt();
        scanner.close();

        int a = 0, b = 1;
        System.out.print(a + " " + b);

        for (int i = 3; i <= n; i++) {
            int nextTerm = a + b;
            System.out.print(" " + nextTerm);
            a = b;
            b = nextTerm;
        }
    }
}
```

Output-



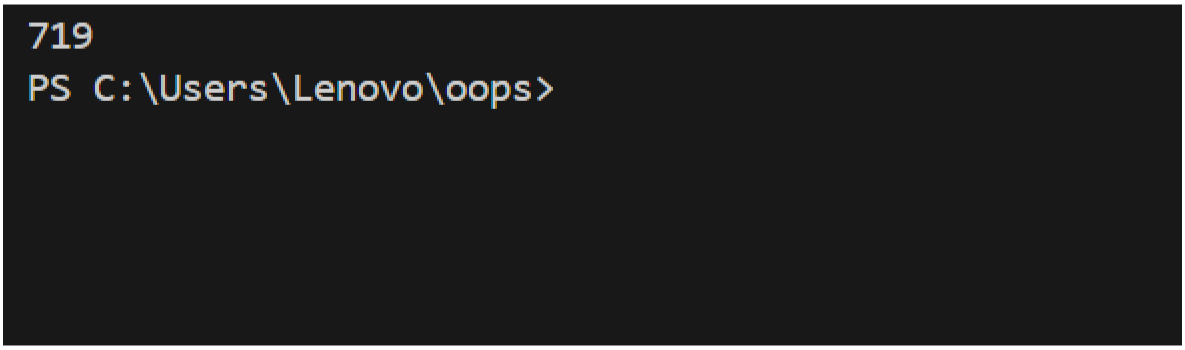
```
Enter the number of terms for Fibonacci series:
10
0 1 1 2 3 5 8 13 21 34
PS C:\Users\Lenovo\oops>
```

11. Write a Java program to reverse a number.

Input-

```
public class reverse {
   public static void main(String[] args) {
      int a = 917;
      int rev  = 0;
      while (a != 0) {
         int n = a%10;
          rev = rev*10 + n;
          a = a/10;
      }
      System.out.println(rev);
   }
}
```

Output-



```
719
PS C:\Users\Lenovo\oops>
```

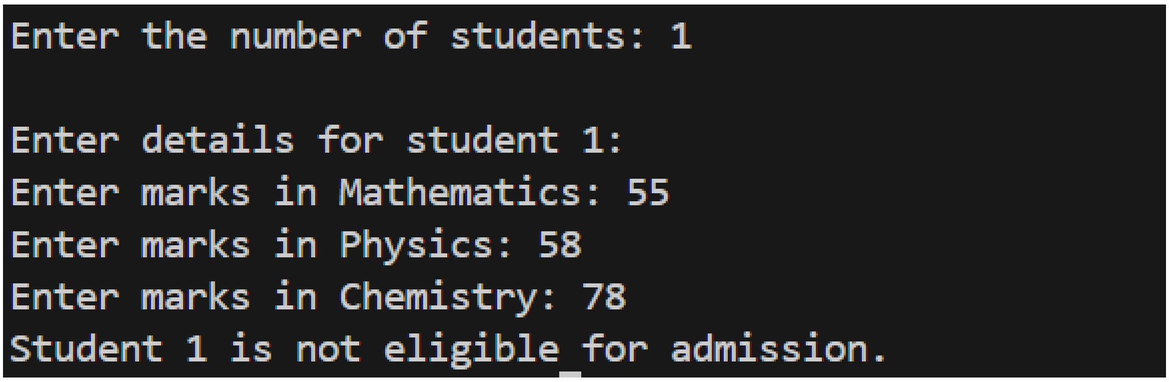
12. Admission to a professional course is subject to the following conditions:
(a) marks in Mathematics >= 60 (b) marks in Physics >=50 (c) marks in Chemistry >=40 (d) Total in all 3 subjects >=200
(Or) Total in Maths & Physics>=150 Given the marks in the 3 subjects of n (user input) students, write a program to processthe applications to list the eligible candidates.

Input-

```
import java.util.Scanner;

public class AdmissionEligibility {

   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter the number of students: ");
      int n = scanner.nextInt();

      for (int i = 1; i <= n; i++) {
         System.out.println("\nEnter details for student " + i + ":");

         System.out.print("Enter marks in Mathematics: ");
         int mathMarks = scanner.nextInt();

         System.out.print("Enter marks in Physics: ");
         int physicsMarks = scanner.nextInt();

         System.out.print("Enter marks in Chemistry: ");
         int chemistryMarks = scanner.nextInt();

         boolean isEligible = checkEligibility(mathMarks, physicsMarks, chemistryMarks);

         if (isEligible) {
            System.out.println("Student " + i + " is eligible for admission.");
         } else {
            System.out.println("Student " + i + " is not eligible for admission.");
         }
      }

      scanner.close();
   }

   public static boolean checkEligibility(int mathMarks, int physicsMarks, int
chemistryMarks) {
      int totalMarks = mathMarks + physicsMarks + chemistryMarks;
      int mathAndPhysicsTotal = mathMarks + physicsMarks;
```

```
        return (mathMarks >= 60 && physicsMarks >= 50 && chemistryMarks >= 40 &&
totalMarks >= 200)
                || (mathAndPhysicsTotal >= 150);
    }
}
```

Output-



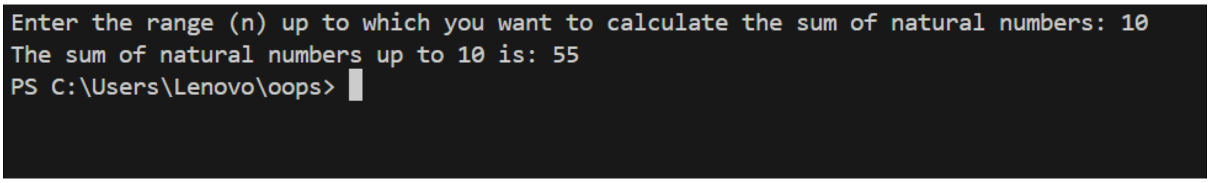
```
Enter the number of students: 1

Enter details for student 1:
Enter marks in Mathematics: 55
Enter marks in Physics: 58
Enter marks in Chemistry: 78
Student 1 is not eligible for admission.
```

13. Write a Java program to calculate the sum of natural numbers up to a certain range.

Input-

```
import java.util.Scanner;

public class SumOfNaturalNumbers {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the range (n) up to which you want to calculate the sum of natural numbers: ");
        int n = scanner.nextInt();


        int sum = (n * (n + 1)) / 2;

        System.out.println("The sum of natural numbers up to " + n + " is: " + sum);

        scanner.close();
    }
}
```

Output –



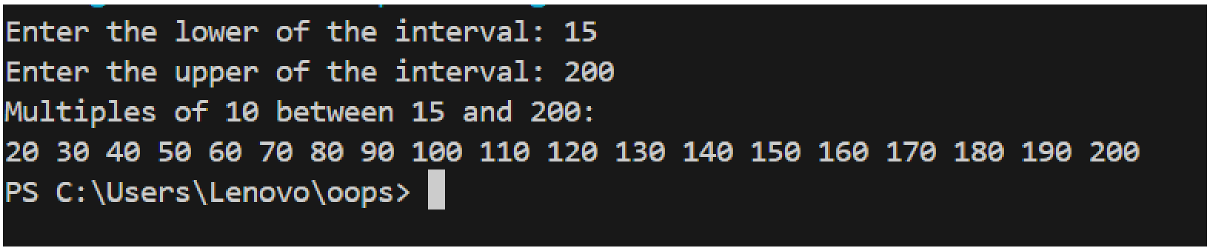
```
Enter the range (n) up to which you want to calculate the sum of natural numbers: 10
The sum of natural numbers up to 10 is: 55
PS C:\Users\Lenovo\oops> 
```

14. Write a Java program to print all multiple of 10 between a given interval.

Input-

```
import java.util.Scanner;

public class Q14 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the lower of the interval: ");
        int lower = scanner.nextInt();

        System.out.print("Enter the upper of the interval: ");
        int upper = scanner.nextInt();

        System.out.println("Multiples of 10 between " + lower + " and " + upper + ":");

        int start = (lower % 10 == 0) ? lower : lower + (10 - lower % 10);

        for (int i = start; i <= upper; i += 10) {
            System.out.print(i + " ");
        }

        scanner.close();
    }
}
```

Output-



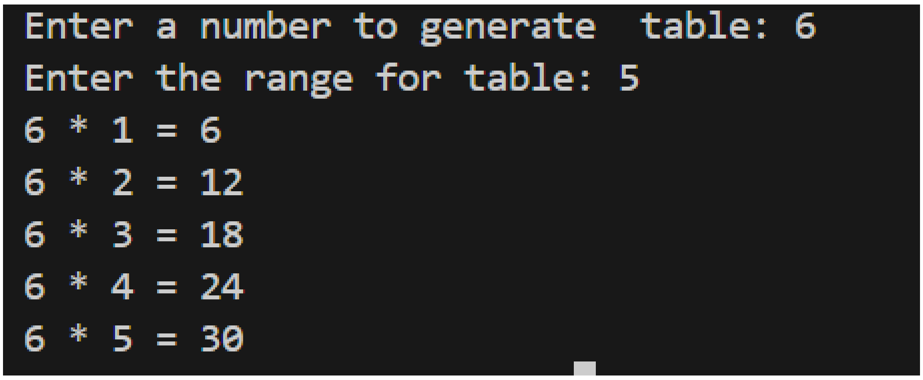
```
Enter the lower of the interval: 15
Enter the upper of the interval: 200
Multiples of 10 between 15 and 200:
20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
PS C:\Users\Lenovo\oops>
```

15. Write a Java program to generate multiplication table.

Input-

```
import java.util.Scanner;

public class Q15{

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number to generate  table: ");
        int number = scanner.nextInt();

        System.out.print("Enter the range for table: ");
        int range = scanner.nextInt();

        for (int i = 1; i <= range; i++) {
            System.out.println(number + " * " + i + " = " + (number * i));
        }

        scanner.close();
    }
}
```

Output-



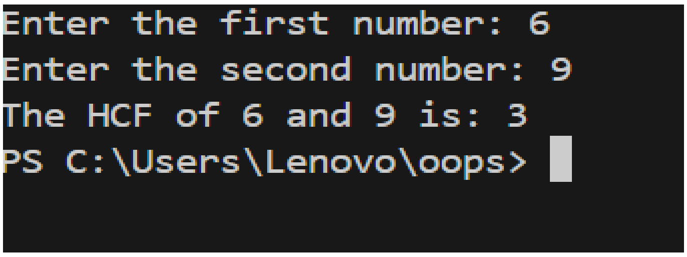
```
Enter a number to generate  table: 6
Enter the range for table: 5
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
```

16. Write a Java program to find HCF of two Numbers.

Input-

```
import java.util.Scanner;

public class Q16 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the first number: ");
        int num1 = scanner.nextInt();

        System.out.print("Enter the second number: ");
        int num2 = scanner.nextInt();

        int hcf = findHCF(num1, num2);

        System.out.println("The HCF of " + num1 + " and " + num2 + " is: " + hcf);

        scanner.close();
    }

    public static int findHCF(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
}
```

Output-



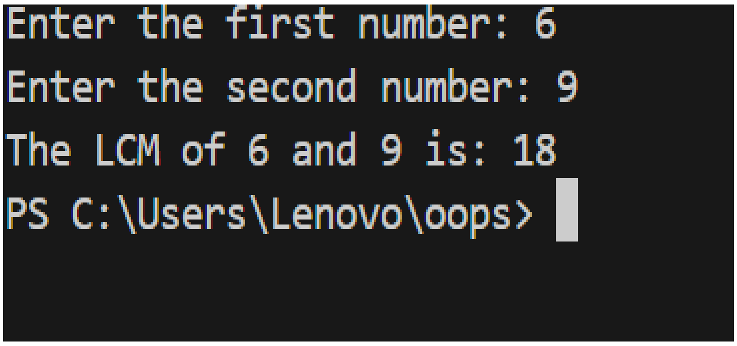
```
Enter the first number: 6
Enter the second number: 9
The HCF of 6 and 9 is: 3
PS C:\Users\Lenovo\oops>
```

17. Write a Java program to find LCM of two Numbers.

Input-

```
import java.util.Scanner;

public class Q17 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the first number: ");
        int num1 = scanner.nextInt();

        System.out.print("Enter the second number: ");
        int num2 = scanner.nextInt();

        int lcm = findLCM(num1, num2);

        System.out.println("The LCM of " + num1 + " and " + num2 + " is: " + lcm);

        scanner.close();
    }

    public static int findLCM(int a, int b) {
        int hcf = findHCF(a, b);
        return (a * b) / hcf;
    }

    public static int findHCF(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
}
```

OUTPUT-

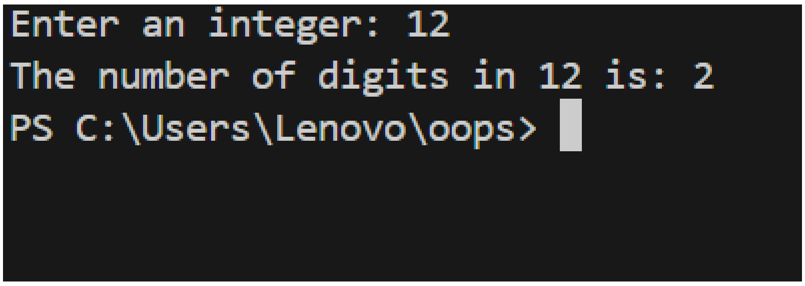


```
Enter the first number: 6
Enter the second number: 9
The LCM of 6 and 9 is: 18
PS C:\Users\Lenovo\oops>
```

18. Write a Java program to count the number of digits of an integer.

Input-

```
import java.util.Scanner;

public class Q18 {

   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter an integer: ");
      int number = scanner.nextInt();

      int digitCount = countDigits(number);

      System.out.println("The number of digits in " + number + " is: " + digitCount);

      scanner.close();
   }

   public static int countDigits(int number) {

      number = Math.abs(number);

      if (number == 0) {
         return 1;
      }

      int count = 0;
      while (number > 0) {
         number = number / 10;
         count++;
      }

      return count;
   }
}
```

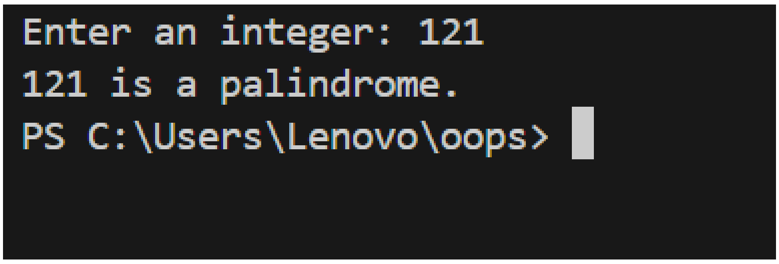


```
Enter an integer: 12
The number of digits in 12 is: 2
PS C:\Users\Lenovo\oops>
```

19. Write a Java program to check whether a number is palindrome or not.

Input-

```
import java.util.Scanner;

public class Q19 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();

        if (isPalindrome(number)) {
            System.out.println(number + " is a palindrome.");
        } else {
            System.out.println(number + " is not a palindrome.");
        }

        scanner.close();
    }

    public static boolean isPalindrome(int number) {
        int originalNumber = number;
        int reversedNumber = 0;

        while (number != 0) {
            int digit = number % 10;
            reversedNumber = reversedNumber * 10 + digit;
            number = number / 10;
        }

        return originalNumber == reversedNumber;
    }
}
```

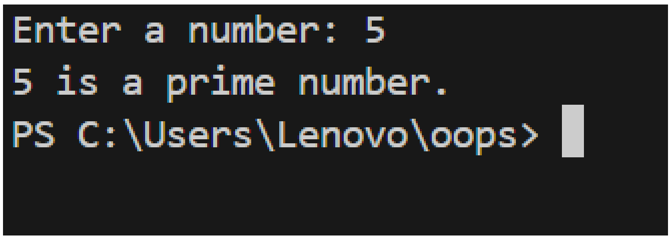


```
Enter an integer: 121
121 is a palindrome.
PS C:\Users\Lenovo\oops>
```

20. Write a Java program to check whether a number is prime or not

Input-

```
import java.util.Scanner;

public class Q205 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        if (isPrime(number)) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }

        scanner.close();
    }

    public static boolean isPrime(int number) {
        if (number <= 1) {
            return false;
        }

        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false;
            }
        }

        return true;
    }
}
```



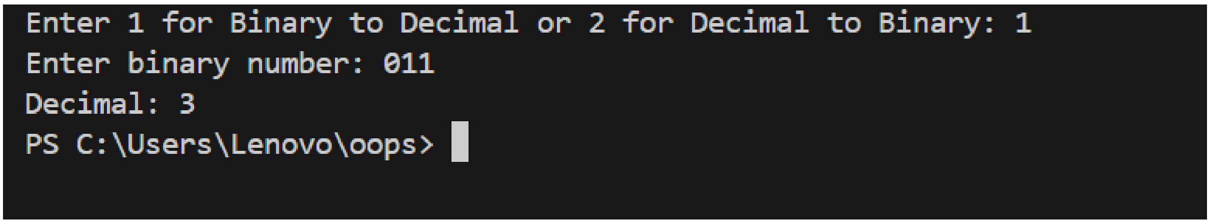
```
Enter a number: 5
5 is a prime number.
PS C:\Users\Lenovo\oops>
```

21. Write a Java program to convert a Binary Number to Decimal and Decimal to Binary.

Input-

```
import java.util.Scanner;

public class Q21 {

  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter 1 for Binary to Decimal or 2 for Decimal to Binary: ");
    int choice = scanner.nextInt();

    if (choice == 1) {
      System.out.print("Enter binary number: ");
      String binary = scanner.next();
      System.out.println("Decimal: " + Integer.parseInt(binary, 2));
    } else if (choice == 2) {
      System.out.print("Enter decimal number: ");
      int decimal = scanner.nextInt();
      System.out.println("Binary: " + Integer.toBinaryString(decimal));
    } else {
      System.out.println("Invalid choice");
    }

    scanner.close();
  }
}
```

Output-



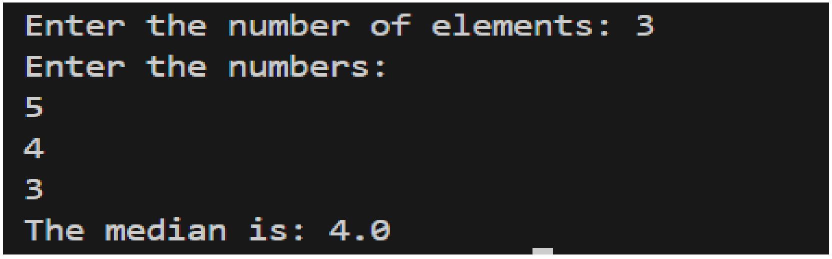
```
Enter 1 for Binary to Decimal or 2 for Decimal to Binary: 1
Enter binary number: 011
Decimal: 3
PS C:\Users\Lenovo\oops>
```

22. Write a Java program to find median of a set of numbers.

Input-

```
import java.util.Arrays;
import java.util.Scanner;

public class Q22 {

   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter the number of elements: ");
      int n = scanner.nextInt();

      int[] numbers = new int[n];

      System.out.println("Enter the numbers:");
      for (int i = 0; i < n; i++) {
         numbers[i] = scanner.nextInt();
      }

      double median = findMedian(numbers);
      System.out.println("The median is: " + median);

      scanner.close();
   }

   public static double findMedian(int[] numbers) {
      Arrays.sort(numbers);

      int n = numbers.length;
      if (n % 2 == 0) {

         return (numbers[n / 2 - 1] + numbers[n / 2]) / 2.0;
      } else {

         return numbers[n / 2];
      }
   }
}
```

Output-



```
Enter the number of elements: 3
Enter the numbers:
5
4
3
The median is: 4.0
```

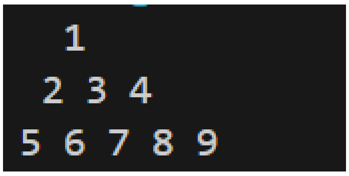
23. Write Java programs for the patterns given bellow:

(a) 1
2 3 4
5 6 7 8 9

Input-

```
public class Q23a {

    public static void main(String[] args) {
        int number = 1;
        for (int i = 1; i <= 3; i++) {

            for (int j = 1; j <= 3 - i; j++) {
                System.out.print(" ");
            }

            for (int j = 1; j <= 2 * i - 1; j++) {
                System.out.print(number + " ");
                number++;
            }

            System.out.println();
        }
    }
}
```

Output-



```
  1
 2 3 4
5 6 7 8 9
```

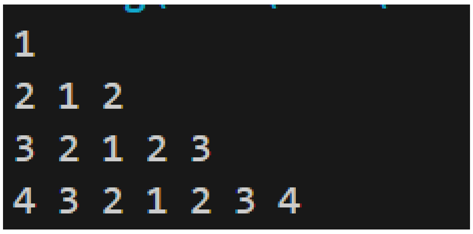
(b)

```
      1
    2 1 2
  3 2 1 2 3
4 3 2 1 2 3 4
```

Input-

```
public class Q23b {


    public static void main(String[] args) {
        for (int i = 1; i <= 4; i++) {
            for (int j = i; j >= 1; j--) {
                System.out.print(j + " ");
            }
            for (int j = 2; j <= i; j++) {
                System.out.print(j + " ");
            }
            System.out.println();
        }
    }
}
```

Output-



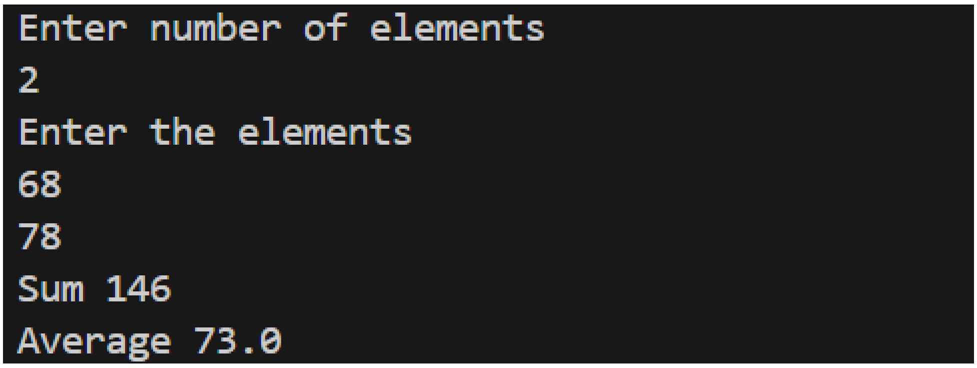
```
1
2 1 2
3 2 1 2 3
4 3 2 1 2 3 4
```

24. Write a Java program to calculate Sum & Average of an integer array.

Input-

```
import java.util.Scanner;

public class Q24 {

    public static void main(String[] args) {
     Scanner sc = new Scanner(System.in);
     System.out.println("Enter number of elements ");
     int n = sc.nextInt();
     int[] arr = new int[n];

     System.out.println("Enter the elements ");
     for (int i = 0; i < n; i++) {
        arr[i] = sc.nextInt();
     }

     int sum = 0;
     for (int i = 0; i < n; i++) {
        sum += arr[i];
     }

     double average = (double) sum / n;
     System.out.println("Sum " + sum);
     System.out.println("Average " + average);
   }
}
```

Output-



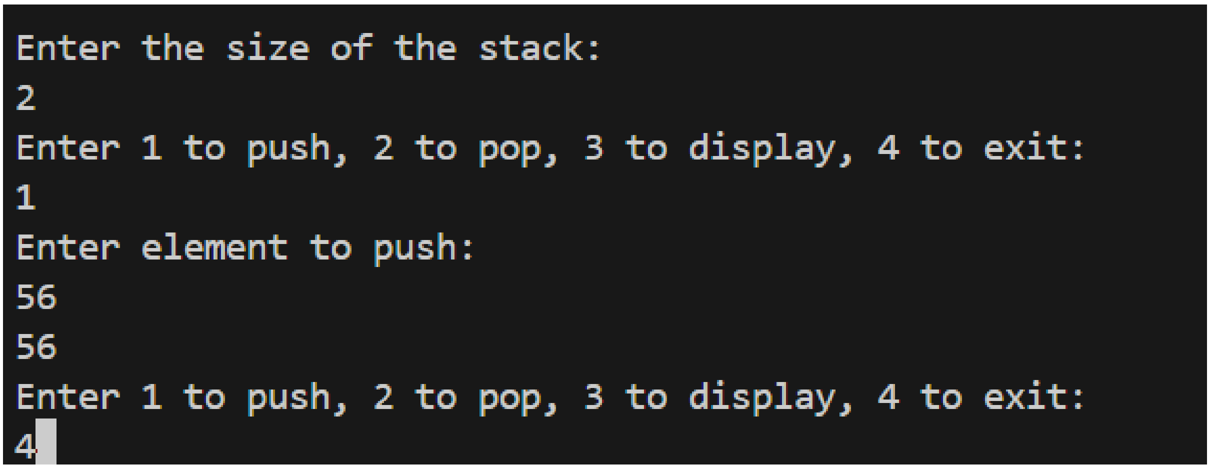
```
Enter number of elements
2
Enter the elements
68
78
Sum 146
Average 73.0
```

25. Write a Java program to implement stack using array

Input-

```
import java.util.Scanner;

public class Q25 {
    static int top = -1;
    static int[] stack;

    public static boolean isEmpty() {
        return top == -1;
    }

    public static void push(int element) {
        if (top == stack.length - 1) {
            System.out.println("Stack overflow");
        } else {
            stack[++top] = element;
            System.out.println(element);
        }
    }


    public static void pop() {
        if (isEmpty()) {
            System.out.println("Stack underflow");
        } else {
            System.out.println(stack[top--]);
        }
    }

    public static void display() {
        if (isEmpty()) {
            System.out.println("Stack is empty");
        } else {
            System.out.println("Stack elements: ");
            for (int i = top; i >= 0; i--) {
                System.out.println(stack[i]);
            }
        }
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the size of the stack: ");
        int size = sc.nextInt();
        stack = new int[size];
        while (true) {
            System.out.println("Enter 1 to push, 2 to pop, 3 to display, 4 to exit: ");
            int choice = sc.nextInt();

            if (choice == 1) {
                System.out.println("Enter element to push: ");
                int element = sc.nextInt();
                push(element);
            } else if (choice == 2) {
                pop();
            } else if (choice == 3) {
                display();
            } else if (choice == 4) {
                break;
            } else {
                System.out.println("Invalid choice, please try again.");
            }
        }
    }

}
```

Output-



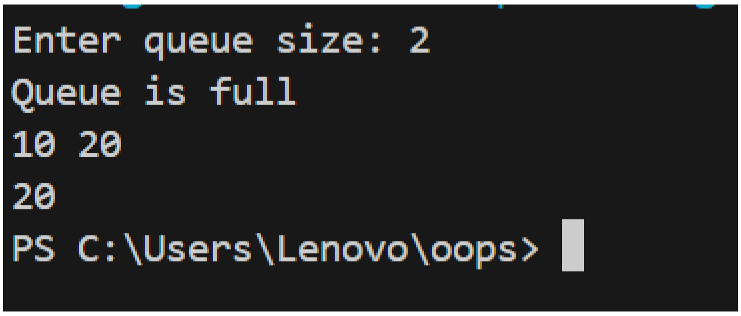
```
Enter the size of the stack:
2
Enter 1 to push, 2 to pop, 3 to display, 4 to exit:
1
Enter element to push:
56
56
Enter 1 to push, 2 to pop, 3 to display, 4 to exit:
4
```

26. Write a Java program to implement Queue using array.

Input-

```
import java.util.Scanner;

public class Queue {
    int[] queue;
    int front, rear, size;

    public Queue(int capacity) {
        queue = new int[capacity];
        front = rear = -1;
        size = capacity;
    }

    boolean isEmpty() { return front == -1; }

    boolean isFull() { return rear == size - 1; }

    void enqueue(int value) {
        if (isFull()) System.out.println("Queue is full");
        else {
            if (front == -1) front = 0;
            queue[++rear] = value;
        }
    }

    void dequeue() {
        if (isEmpty()) System.out.println("Queue is empty");
        else front++;
    }

    void display() {
        if (isEmpty()) System.out.println("Queue is empty");
        else {
            for (int i = front; i <= rear; i++) System.out.print(queue[i] + " ");
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter queue size: ");
        int size = scanner.nextInt();
        Queue q = new Queue(size);

        q.enqueue(10);
        q.enqueue(20);
        q.enqueue(30);

        q.display();

        q.dequeue();
        q.display();

        scanner.close();
    }
}
```

Output-



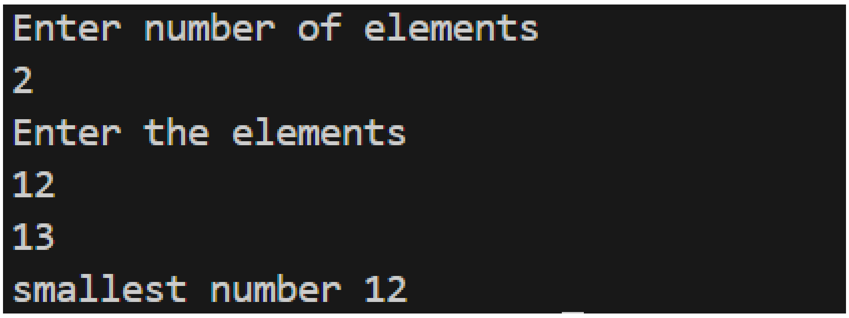
```
Enter queue size: 2
Queue is full
10 20
20
PS C:\Users\Lenovo\oops>
```

27. Write a Java program to enter n elements in an array and find smallest number among them.

Input-

```
import java.util.*;

public class Q27 {


    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of elements ");
        int n = sc.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

       Arrays.sort(arr);
       System.out.println("smallest number " + arr[0]);

    }
}
```

Output-



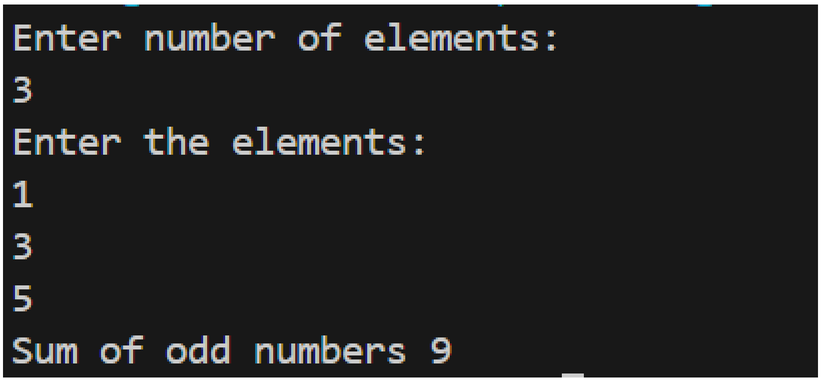
```
Enter number of elements
2
Enter the elements
12
13
smallest number 12
```

28. Write Java program to find the sum of all odd numbers in a array.

Input-

```
import java.util.*;
public class Q28 {


    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        int sum = 0;
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] % 2 != 0) {
                sum = sum + arr[i];
            }
        }

        System.out.println("Sum of odd numbers " + sum);
    }

}
```

Output-



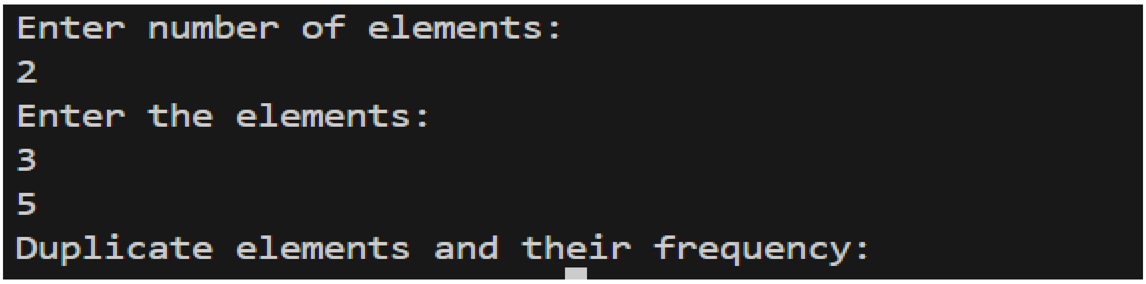
```
Enter number of elements:
3
Enter the elements:
1
3
5
Sum of odd numbers 9
```

29. Write a Java program to find duplicate elements in a 1D array and find their frequency of occurrence

Input-

```
import java.util.*;

public class Q29 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        System.out.println("Duplicate elements and their frequency: ");

        for (int i = 0; i < n; i++) {
            int count = 1;
            if (arr[i] != -1) {
                for (int j = i + 1; j < n; j++) {
                    if (arr[i] == arr[j]) {
                        count++;
                        arr[j] = -1;
                    }
                }
                if (count > 1) {
                    System.out.println(arr[i] + " occurs " + count + " times");
                }
            }
        }
    }
}
```

Output-



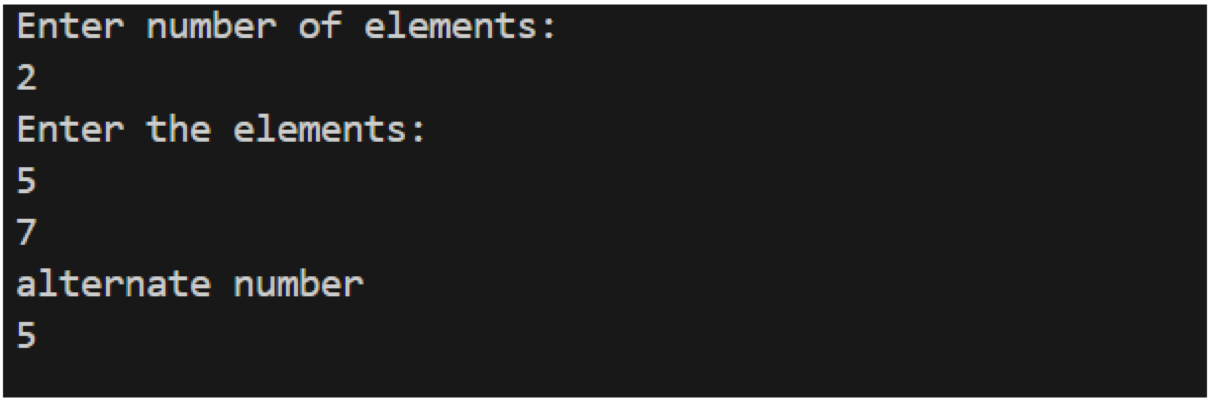
```
Enter number of elements:
2
Enter the elements:
3
5
Duplicate elements and their frequency:
```

30. Write a Java program to print every alternate number of a given array.

Input-

```
import java.util.*;

public class Q30 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.println("alternate number");
        for (int i = 0; i < arr.length; i+= 2) {
            System.out.println(arr[i] + " ");
        }
        System.out.println();
    }
}
```

Output-



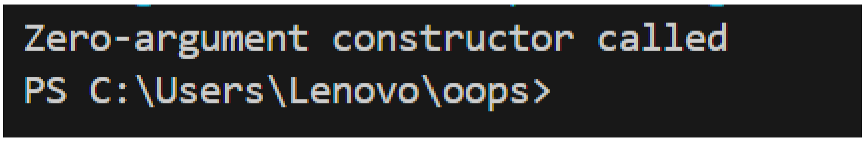
```
Enter number of elements:
2
Enter the elements:
5
7
alternate number
5
```

31. Write a Java program to show 0-arguments constructor.

Input-

```
class MyClass {
    public MyClass() {
      System.out.println("Zero-argument constructor called");
    }
  }
  public class Q31 {

    public static void main(String[] args) {
      MyClass obj = new MyClass();
    }
}
```

Output-



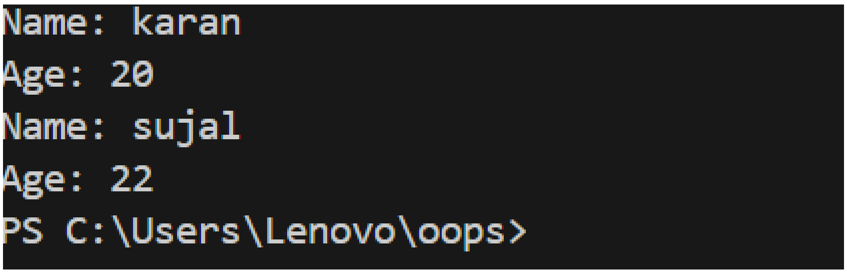
```
Zero-argument constructor called
PS C:\Users\Lenovo\oops>
```

32. Write a Java program to show parameterized constructor.

Input-

```
class Student {
    String name;
    int age;

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


    public void display() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}

public class Q32 {
    public static void main(String[] args) {

        Student student1 = new Student("karan", 20);
        Student student2 = new Student("sujal", 22);


        student1.display();
        student2.display();
    }
}
```

Output-



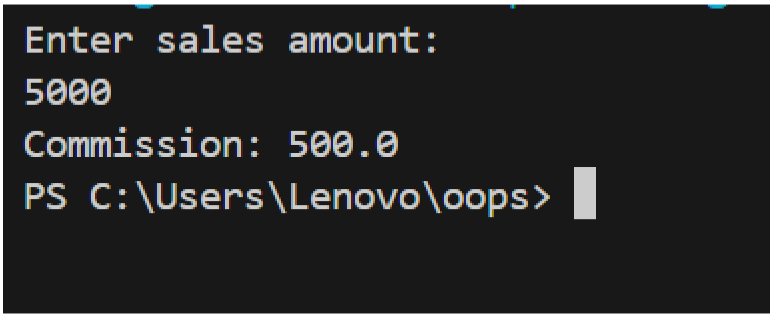
```
Name: karan
Age: 20
Name: sujal
Age: 22
PS C:\Users\Lenovo\oops>
```

33. Write a class, Commission, which has an instance variable, sales; an appropriate constructor; and a method, commission() that returns the commission.Now write a demo class to test the Commission class by reading a sale from the user, using it to create a Commission object after validating that the value is not negative. Finally, call the commission() method to get and print the commission. If the sales are negative, your demo should print the message “Invalid Input”.

Input-

```
import java.util.*;

class Commission {
    private double sales;

    public Commission(double sales) {
        if (sales < 0) {
            System.out.println("Invalid Input");
            this.sales = 0;
        } else {
            this.sales = sales;
        }
    }

    public double calculateCommission() {
        return sales * 0.1;
    }
}

public class Q33 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter sales amount: ");
        double sales = sc.nextDouble();

        Commission commissionObj = new Commission(sales);
        if (sales >= 0) {
            System.out.println("Commission: " + commissionObj.calculateCommission());
        }
    }
}
```

Output-



```
Enter sales amount:
5000
Commission: 500.0
PS C:\Users\Lenovo\oops>
```

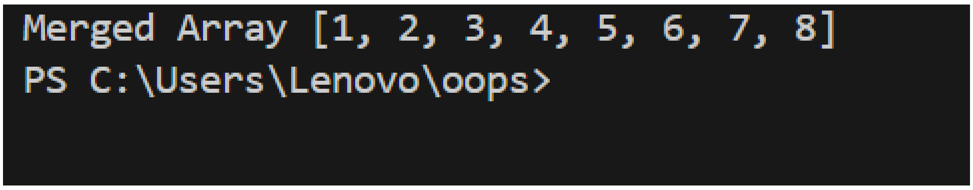
ASSIGNMENT 2

1. Given are two one-dimensional arrays A & B, which are sorted in ascending order. Write a Java program to merge them into single sorted array C that contains every item from arrays A & B, in ascending order.

Input-

```
import java.util.*;

public class q1 {

    public static void main(String[] args) {
        int[] A = { 1, 3, 5, 7 };
        int[] B = { 2, 4, 6, 8 };

        int[] C = new int[A.length + B.length];
        int i = 0, j = 0, k = 0;

        while (i < A.length && j < B.length) {
            if (A[i] < B[j]) {
                C[k++] = A[i++];
            } else {
                C[k++] = B[j++];
            }
        }

        while (i < A.length) {
            C[k++] = A[i++];
        }

        while (j < B.length) {
            C[k++] = B[j++];
        }

        System.out.println("Merged Array " + Arrays.toString(C));
    }
}
```

Output-



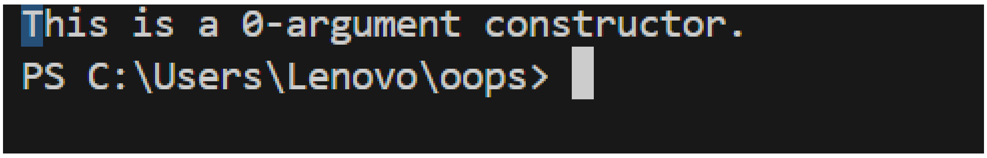
```
Merged Array [1, 2, 3, 4, 5, 6, 7, 8]
PS C:\Users\Lenovo\oops>
```

2. Write a Java program to show 0-arguments constructor.

Input-

```
public class q2 {


   q2() {
      System.out.println("This is a 0-argument constructor.");
   }

   public static void main(String[] args) {
      new q2();
   }
}
```

Output-



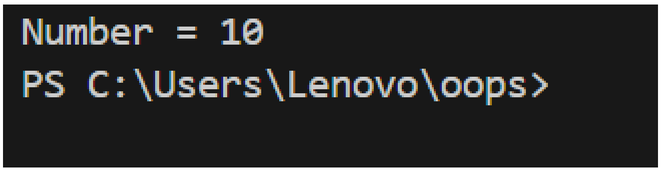
```
This is a 0-argument constructor.
PS C:\Users\Lenovo\oops>
```

3. Write a Java program to show parameterized constructor

Input-

```
public class q3 {


    int number;

    q3(int num) {
        number = num;
    }

    public static void main(String[] args) {
        q3 obj = new q3(10);
        System.out.println("Number = " + obj.number);
    }
}
```

Output –



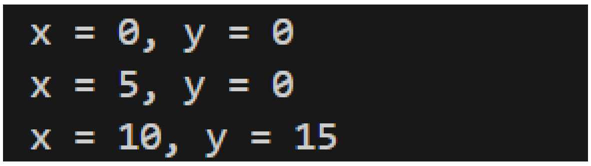
```
Number = 10
PS C:\Users\Lenovo\oops>
```

4. Write a Java program to show constructor overloading.

Input-

```
public class q4 {
    public static void main(String[] args) {
        Example obj1 = new Example();
        Example obj2 = new Example(5);
        Example obj3 = new Example(10, 15);

        obj1.display();
        obj2.display();
        obj3.display();
    }
}
class Example {
    int x;
    int y;

    Example() {
        x = 0;
        y = 0;
    }

    Example(int a) {
        x = a;
        y = 0;
    }

    Example(int a, int b) {
        x = a;
        y = b;
    }

    void display() {
        System.out.println("x = " + x + ", y = " + y);
    }
}
```

Output-



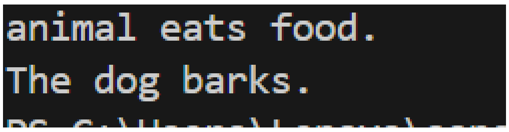
```
x = 0, y = 0
x = 5, y = 0
x = 10, y = 15
```

5. Write a Java program to implement the concept of inheritance.

Input-

```
public class q5 {

   public static void main(String[] args) {
      Dog dog = new Dog();
      dog.eat();
      dog.bark();
   }

}

class Animal {
   void eat() {
      System.out.println("animal eats food.");
   }
}

class Dog extends Animal {
   void bark() {
      System.out.println("The dog barks.");
   }
}
```

Output-



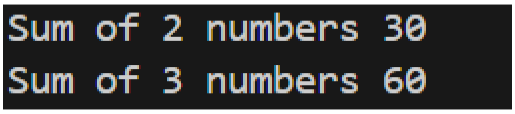
```
animal eats food.
The dog barks.
```

6. Write a Java program to show method overloading.

Input-

```
public class q6 {


   int add(int a, int b) {
      return a + b;
   }

   int add(int a, int b, int c) {
      return a + b + c;
   }

   public static void main(String[] args) {
      q6 obj = new q6();
      System.out.println("Sum of 2 numbers " + obj.add(10, 20));
      System.out.println("Sum of 3 numbers " + obj.add(10, 20, 30));
   }
}
```

Output-



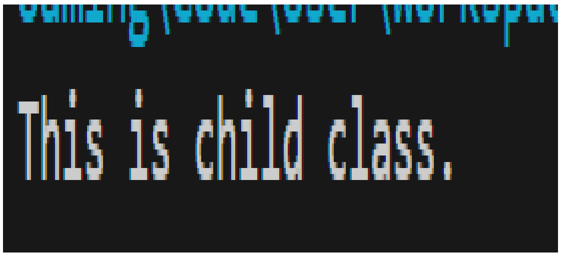
```
Sum of 2 numbers 30
Sum of 3 numbers 60
```

7. Write a Java program to show method overriding.

Input-

```
public class q7 {
        public static void main(String[] args) {
        Child child = new Child();
        child.display();
    }

}

class Parent {
    void display() {
        System.out.println("This is parent class.");
    }
}

class Child extends Parent {
    @Override
    void display() {
        System.out.println("This is child class.");
    }
}
```

Output-



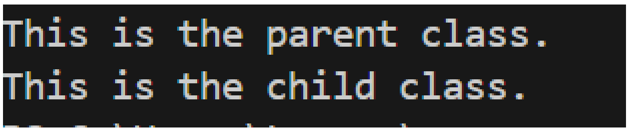
```
This is child class.
```

8. Write a Java program to show method hiding.

Input-

```
public class q8 {

    class Parent {
        static void show() {
            System.out.println("This is the parent class.");
        }
    }

    class Child extends Parent {
        static void show() {
            System.out.println("This is the child class.");
        }
    }

    public static void main(String[] args) {
        Parent.show();
        Child.show();
    }
}
```

Output-



```
This is the parent class.
This is the child class.
```

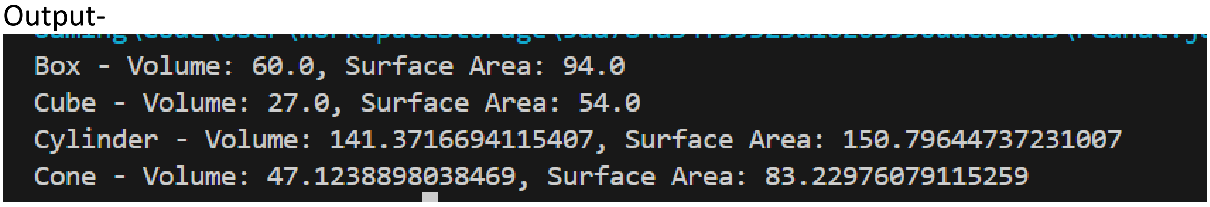
9. Create a general class ThreeDObject and derive the classes Box, Cube, Cylinder and Cone from it. The class ThreeDObject has methods wholeSurfaceArea ( ) and volume (). Override these two methods in each of the derived classes to calculate the volume and whole surface area of each type of three-dimensional objects. The dimensions of the objects are to be taken from the users and passed through the respective constructors of each derived class. Write a main method to test these classes.

Input-

```
public class q99 {
   public static void main(String[] args) {
      Box box = new Box(3, 4, 5);
      Cube cube = new Cube(3);
      Cylinder cylinder = new Cylinder(3, 5);
      Cone cone = new Cone(3, 5);

      System.out.println("Box - Volume: " + box.volume() + ", Surface Area: " +
box.wholeSurfaceArea());
      System.out.println("Cube - Volume: " + cube.volume() + ", Surface Area: " +
cube.wholeSurfaceArea());
      System.out
            .println("Cylinder - Volume: " + cylinder.volume() + ", Surface Area: " +
cylinder.wholeSurfaceArea());
      System.out.println("Cone - Volume: " + cone.volume() + ", Surface Area: " +
cone.wholeSurfaceArea());
   }

}

abstract class ThreeDObject {
   abstract double wholeSurfaceArea();

   abstract double volume();
}

class Box extends ThreeDObject {
   double length, width, height;

   Box(double l, double w, double h) {
      length = l;
      width = w;
      height = h;
   }

   double wholeSurfaceArea() {
      return 2 * (length * width + width * height + height * length);
   }
```

```
    double volume() {
        return length * width * height;
    }
}

class Cube extends ThreeDObject {
    double side;

    Cube(double s) {
        side = s;
    }

    double wholeSurfaceArea() {
        return 6 * side * side;
    }

    double volume() {
        return side * side * side;
    }
}

class Cylinder extends ThreeDObject {
    double radius, height;

    Cylinder(double r, double h) {
        radius = r;
        height = h;
    }

    double wholeSurfaceArea() {
        return 2 * Math.PI * radius * (radius + height);
    }

    double volume() {
        return Math.PI * radius * radius * height;
    }
}

class Cone extends ThreeDObject {
    double radius, height;

    Cone(double r, double h) {
        radius = r;
        height = h;
    }

    double wholeSurfaceArea() {
        double slantHeight = Math.sqrt(radius * radius + height * height);
        return Math.PI * radius * (radius + slantHeight);
    }

    double volume() {
        return Math.PI * radius * radius * height / 3;
    }
}
```

Output-



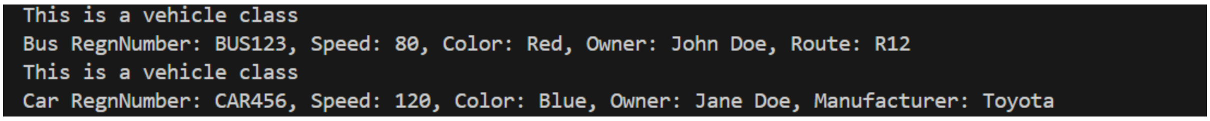
```
Box - Volume: 60.0, Surface Area: 94.0
Cube - Volume: 27.0, Surface Area: 54.0
Cylinder - Volume: 141.3716694115407, Surface Area: 150.79644737231007
Cone - Volume: 47.1238898038469, Surface Area: 83.22976079115259
```

10. Write a program to create a class named Vehicle having protected instance variables regnNumber, speed, color, ownerName and a method showData ( ) to show “This is a vehicle class”. Inherit the Vehicle class into subclasses named Bus and Car having individual private instance variables routeNumber in Bus and manufacturerName in Car and both of them having showData ( ) method showing all details of Bus and Car respectively with content of the super class’s showData ( ) method.

Input-

```
public class q10 {
    public static void main(String[] args) {
        Bus bus = new Bus("BUS123", 80, "Red", "John Doe", "R12");
        Car car = new Car("CAR456", 120, "Blue", "Jane Doe", "Toyota");

        bus.showData();
        car.showData();
    }
}

class Vehicle {
    protected String regnNumber;
    protected int speed;
    protected String color;
    protected String ownerName;

    Vehicle(String regnNumber, int speed, String color, String ownerName) {
        this.regnNumber = regnNumber;
        this.speed = speed;
        this.color = color;
        this.ownerName = ownerName;
    }

    void showData() {
        System.out.println("This is a vehicle class");
    }
}

class Bus extends Vehicle {
    private String routeNumber;

    Bus(String regnNumber, int speed, String color, String ownerName, String routeNumber) {
        super(regnNumber, speed, color, ownerName);
        this.routeNumber = routeNumber;
    }

    void showData() {
        super.showData();
        System.out.println("Bus RegnNumber: " + regnNumber + ", Speed: " + speed + ", Color: 
" + color + ", Owner: "
                + ownerName + ", Route: " + routeNumber);
    }
}

class Car extends Vehicle {
    private String manufacturerName;

    Car(String regnNumber, int speed, String color, String ownerName, String 
manufacturerName) {
        super(regnNumber, speed, color, ownerName);
        this.manufacturerName = manufacturerName;
    }

    void showData() {
        super.showData();
        System.out.println("Car RegnNumber: " + regnNumber + ", Speed: " + speed + ", Color: " 
+ color + ", Owner: "
                + ownerName + ", Manufacturer: " + manufacturerName);
    }
}
```

Output-



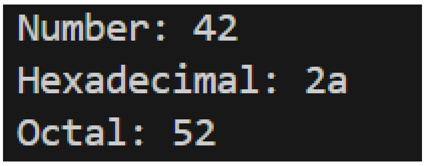
```
This is a vehicle class
Bus RegnNumber: BUS123, Speed: 80, Color: Red, Owner: John Doe, Route: R12
This is a vehicle class
Car RegnNumber: CAR456, Speed: 120, Color: Blue, Owner: Jane Doe, Manufacturer: Toyota
```

11. Write a Java program which creates a base class Num and contains an integer number along with a method shownum() which displays the number. Now create a derived class HexNum which inherits Num and overrides shownum() which displays the hexadecimal value and octal value of the number. Demonstrate the working of the classes.

Input-

```
public class q11 {
    public static void main(String[] args) {
        Num num = new Num(42);
        HexNum hexNum = new HexNum(42);

        num.showNum();
        hexNum.showNum();
    }
}

class Num {
    int number;

    Num(int number) {
        this.number = number;
    }

    void showNum() {
        System.out.println("Number: " + number);
    }
}

class HexNum extends Num {
    HexNum(int number) {
        super(number);
    }

    void showNum() {
        System.out.println("Hexadecimal: " + Integer.toHexString(number));
        System.out.println("Octal: " + Integer.toOctalString(number));
    }
}
```

Output-



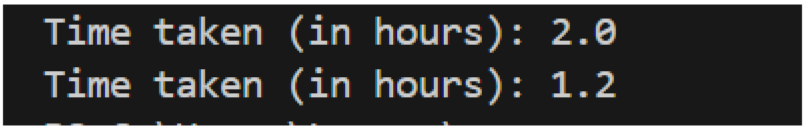
```
Number: 42
Hexadecimal: 2a
Octal: 52
```

12. Create a base class Distance which stores the distance between two locations in miles and a method travelTime(). The method prints the time taken to cover the distance when the speed is 60 miles per hour. Now in a derived class DistanceMKS, override travelTime() so that it prints the time assuming the distance is in kilometers and the speed is 100 km per second. Demonstrate the working of the classes.

Input-

```
public class q12 {
   public static void main(String[] args) {
      Distance dist = new Distance(120);
      DistanceMKS distMKS = new DistanceMKS(120);

      dist.travelTime();
      distMKS.travelTime();
   }
}

class Distance {
   double miles;

   Distance(double miles) {
      this.miles = miles;
   }

   void travelTime() {
      System.out.println("Time taken (in hours): " + (miles / 60));
   }
}

class DistanceMKS extends Distance {
   DistanceMKS(double kilometers) {
      super(kilometers);
   }

   void travelTime() {
      System.out.println("Time taken (in hours): " + (miles / 100));
   }
}
```

Output-



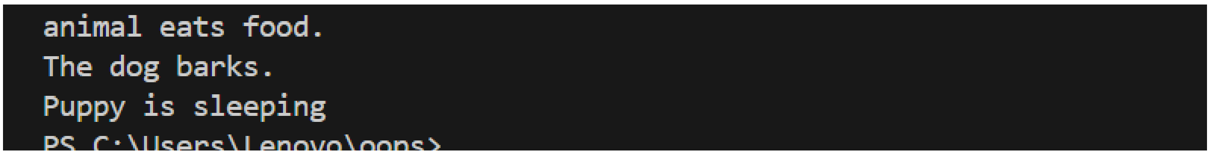
```
Time taken (in hours): 2.0
Time taken (in hours): 1.2
```

13. Write a Java program to explain “multilevel inheritance.

Input-

```
class Animal {
    void eat() {
        System.out.println("Animal is eating");
    }
}

class Dog extends Animal {
    void bark() {
        System.out.println("Dog is barking");
    }
}

class Puppy extends Dog {
    void sleep() {
        System.out.println("Puppy is sleeping");
    }
}

public class q13 {
    public static void main(String[] args) {
        Puppy puppy = new Puppy();
        puppy.eat();
        puppy.bark();
        puppy.sleep();
    }
}
```

Output-



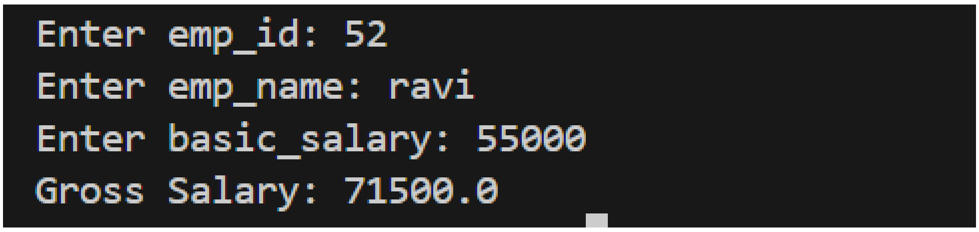
```
animal eats food.
The dog barks.
Puppy is sleeping
PS C:\Users\Lenovo\oops>
```

14. Write a program to define a class Employee to accept emp_id, emp _name, basic_salary from the user and display the gross_salary.

Input-

```
import java.util.*;


public class w14 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter emp_id: ");
        int id = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter emp_name: ");
        String name = sc.nextLine();
        System.out.print("Enter basic_salary: ");
        double salary = sc.nextDouble();

        Employee emp = new Employee(id, name, salary);
        emp.displayGrossSalary();
    }
}

class Employee {
    int emp_id;
    String emp_name;
    double basic_salary;

    Employee(int id, String name, double salary) {
        emp_id = id;
        emp_name = name;
        basic_salary = salary;
    }

    void displayGrossSalary() {
        double gross_salary = basic_salary + (basic_salary * 0.2) + (basic_salary * 0.1);
        System.out.println("Gross Salary: " + gross_salary);
    }
}
```

Output-



```
Enter emp_id: 52
Enter emp_name: ravi
Enter basic_salary: 55000
Gross Salary: 71500.0
```

15. Write a program to demonstrate use of 'this' keyword.

Input-

```
public class w15 {
    public static void main(String[] args) {
        DemoThis obj = new DemoThis(10);
        obj.show();
    }
}

class DemoThis {
    int num;

    DemoThis(int num) {
        this.num = num;
    }

    void show() {
        System.out.println("Number: " + this.num);
    }
}
```

Output-

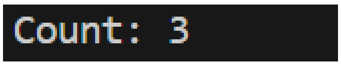
```
Number: 10
```

16. Write a program to demonstrate use of 'this' keyword.

Input-

```
public class w16 {
    public static void main(String[] args) {
        DemoStatic obj1 = new DemoStatic();
        DemoStatic obj2 = new DemoStatic();
        DemoStatic obj3 = new DemoStatic();

        DemoStatic.showCount();
    }
}

class DemoStatic {
    static int count = 0;

    DemoStatic() {
        count++;
    }

    static void showCount() {
        System.out.println("Count: " + count);
    }
}
```

Output-



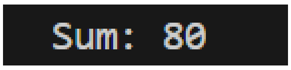
```
Count: 3
```

17. Write program, which finds the sum of numbers formed by consecutive digits.
Input : 2415
output : 24+41+15=80.

Input-

```
public class w17 {
    public static void main(String[] args) {
        int num = 2415;
        String str = Integer.toString(num);
        int sum = 0;

        for (int i = 0; i < str.length() - 1; i++) {
            int consecutiveNum = Integer.parseInt(str.substring(i, i + 2));
            sum += consecutiveNum;
        }

        System.out.println("Sum: " + sum);
    }
}
```

Output-



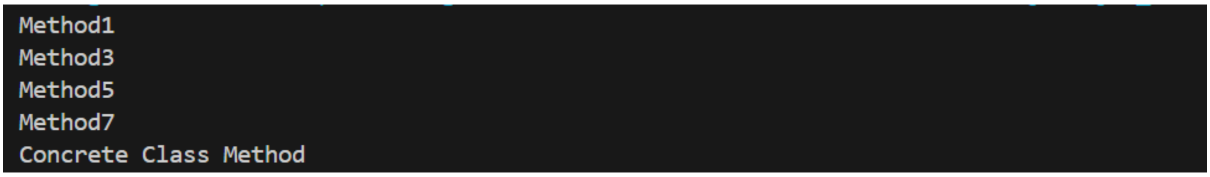
```
Sum: 80
```

18. Create three interfaces, each with two methods. Inherit a new interface from the three, adding a new method. Create a class by implementing the new interface and also inheriting from a concrete class. Now write four methods, each of which takes one of the four interfaces as an argument. In main ( ), create an object of your class and pass it to each of the methods

Input-

```
public class w18 {
    public static void main(String[] args) {
        MultiInterfaceClass obj = new MultiInterfaceClass();
        obj.method1();
        obj.method3();
        obj.method5();
        obj.method7();
        obj.concreteMethod();
    }
}

interface Interface1 {
    void method1();
    void method2();
}

interface Interface2 {
    void method3();
    void method4();
}

interface Interface3 {
    void method5();
    void method6();
}

interface CombinedInterface extends Interface1, Interface2, Interface3 {
    void method7();
}

class ConcreteClass {
    void concreteMethod() {
        System.out.println("Concrete Class Method");
    }
}

class MultiInterfaceClass extends ConcreteClass implements CombinedInterface {
    public void method1() {
        System.out.println("Method1");
    }
    public void method2() {
        System.out.println("Method2");
    }

    public void method3() {
        System.out.println("Method3");
    }

    public void method4() {
        System.out.println("Method4");
    }

    public void method5() {
        System.out.println("Method5");
    }

    public void method6() {
        System.out.println("Method6");
    }

    public void method7() {
        System.out.println("Method7");
    }
}
```

Output-



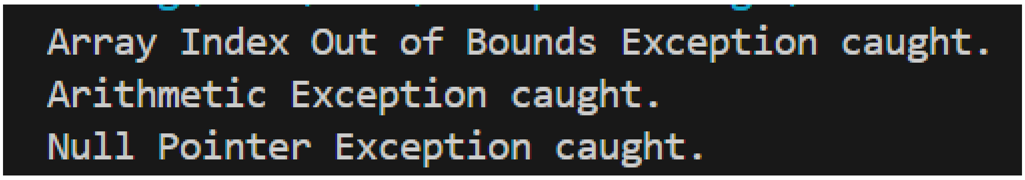
```
Method1
Method3
Method5
Method7
Concrete Class Method
```

19. Write a Java program to show the use of all keywords for exception handling.

Input-

```
public class w19 {
    public static void main(String[] args) {
        try {
            int[] arr = new int[5];
            arr[10] = 100;
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Array Index Out of Bounds Exception caught.");
        }

        try {
            int result = 10 / 0;
        } catch (ArithmeticException e) {
            System.out.println("Arithmetic Exception caught.");
        }

        try {
            String str = null;
            System.out.println(str.length());
        } catch (NullPointerException e) {
            System.out.println("Null Pointer Exception caught.");
        }
    }
}
```

Output-



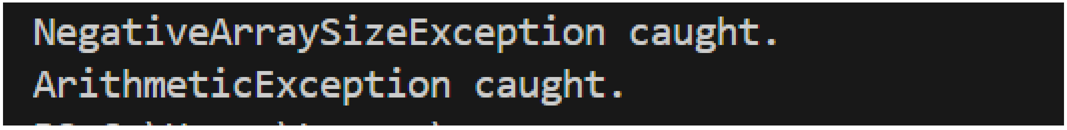
```
Array Index Out of Bounds Exception caught.
Arithmetic Exception caught.
Null Pointer Exception caught.
```

20. Write a Java program using try and catch to generate NegativeArrayIndex Exception and Arithmetic Exception.

Input-

```
public class w20 {
    public static void main(String[] args) {
        try {
            int[] arr = new int[-5];
        } catch (NegativeArraySizeException e) {
            System.out.println("NegativeArraySizeException caught.");
        }

        try {
            int result = 10 / 0;
        } catch (ArithmeticException e) {
            System.out.println("ArithmeticException caught.");
        }
    }
}
```

Output-



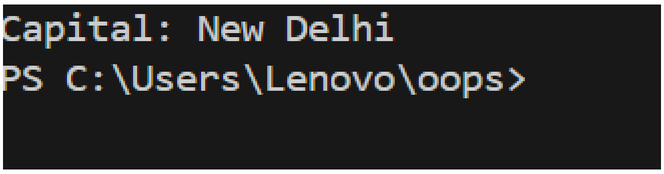
```
NegativeArraySizeException caught.
ArithmeticException caught.
```

21. Write a program that outputs the name of the capital of the country entered at the command line. The program should throw a “NoMatchFoundException” when it fails to print the capital of the country entered at the command line.

Input-

```
public class w21 {
   public static void main(String[] args) {
      String country = "India";  // Example country input

      try {
         String capital = getCapital(country);
         System.out.println("Capital: " + capital);
      } catch (NoMatchFoundException e) {
         System.out.println(e.getMessage());
      }
   }

   public static String getCapital(String country) throws NoMatchFoundException {
      if (country.equalsIgnoreCase("India")) {
         return "New Delhi";
      } else if (country.equalsIgnoreCase("USA")) {
         return "Washington D.C.";
      } else if (country.equalsIgnoreCase("Germany")) {
         return "Berlin";
      } else {
         throw new NoMatchFoundException("No match found for the entered country.");
      }
   }
}
class NoMatchFoundException extends Exception {
   public NoMatchFoundException(String message) {
      super(message);
   }
}
```

Output-



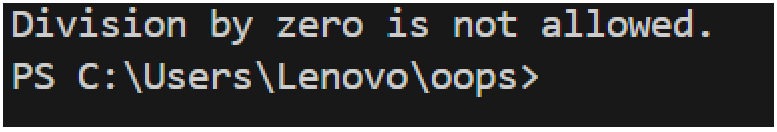
```
Capital: New Delhi
PS C:\Users\Lenovo\oops>
```

22. Write a java program to create an custom Exception that would handle at least 2 kind of Arithmetic Exceptions while calculating a given equation

Input-

```
public class w22 {
    public static void main(String[] args) {
        try {
            int result = calculate(10, 0);
        } catch (CustomArithmeticException e) {
            System.out.println(e.getMessage());
        }
    }

    public static int calculate(int a, int b) throws CustomArithmeticException {
        if (b == 0) {
            throw new CustomArithmeticException("Division by zero is not allowed.");
        }
        return a / b;
    }
}

class CustomArithmeticException extends Exception {
    public CustomArithmeticException(String message) {
        super(message);
    }
}
```

Output-



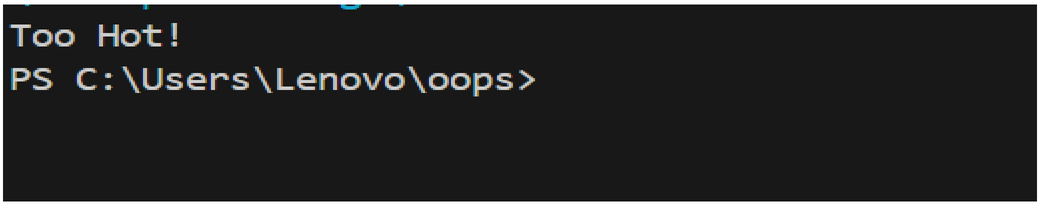
```
Division by zero is not allowed.
PS C:\Users\Lenovo\oops>
```

23. Create two user-defined exceptions named “TooHot” and “TooCold” to check the temperature (in Celsius) given by the user passed through the command line is too hot or too cold.
If temperature > 35, throw exception “TooHot”.
If temperature <5, throw exception “TooCold”.
Otherwise, print “Normal” and convert it to Farenheit.

Input-

```
public class w23 {
    public static void main(String[] args) {
        int temperature = 36; // Example temperature

        try {
            if (temperature > 35) {
                throw new TooHotException("Too Hot!");
            } else if (temperature < 5) {
                throw new TooColdException("Too Cold!");
            } else {
                System.out.println("Normal");
                System.out.println("Fahrenheit: " + (temperature * 9 / 5 + 32));
            }
        } catch (TooHotException | TooColdException e) {
            System.out.println(e.getMessage());
        }
    }
}

class TooHotException extends Exception {
    public TooHotException(String message) {
        super(message);
    }
}

class TooColdException extends Exception {
    public TooColdException(String message) {
        super(message);
    }
}
```

Output-



```
Too Hot!
PS C:\Users\Lenovo\oops>
```

24. Consider an Employee recruitment system that prints the candidate name based on the age criteria. The name and age of the candidate are taken as Input.Create two user-defined exceptions named “TooOlder” and “TooYounger”

If age>45, throw exception “TooOlder”.

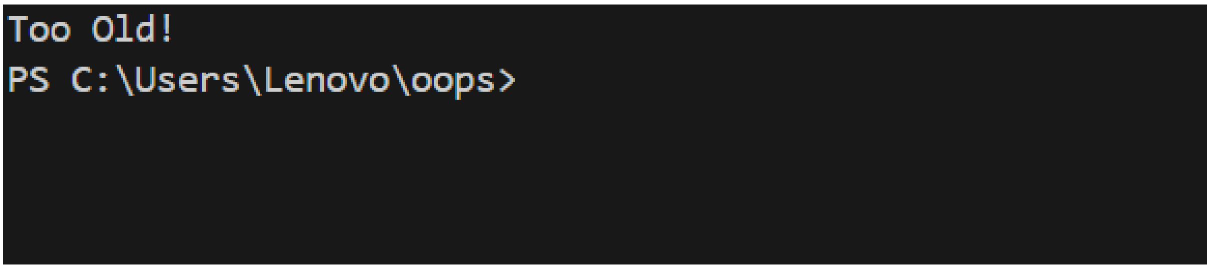
If age<20, throw exception “TooYounger”.

Otherwise, print “Eligible” and print the name of the candidate

Input-

```
public class w24 {
   public static void main(String[] args) {
      int age = 46; // Example age
      String name = "John";

      try {
         if (age > 45) {
            throw new TooOlderException("Too Old!");
         } else if (age < 20) {
            throw new TooYoungerException("Too Young!");
         } else {
            System.out.println("Eligible: " + name);
         }
      } catch (TooOlderException | TooYoungerException e) {
         System.out.println(e.getMessage());
      }
   }
}

class TooOlderException extends Exception {
   public TooOlderException(String message) {
      super(message);
   }
}

class TooYoungerException extends Exception {
   public TooYoungerException(String message) {
      super(message);
   }
}
```

Output-



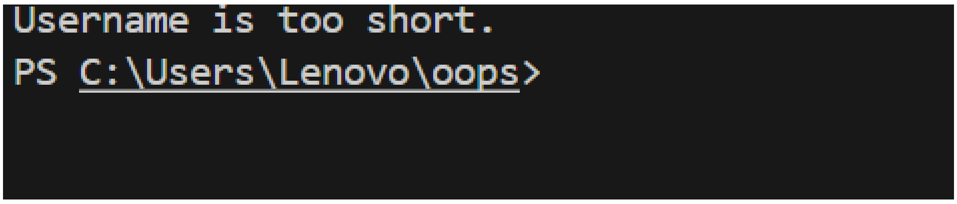
```
Too Old!
PS C:\Users\Lenovo\oops>
```

25. Write a program to raise a user defined exception if username is less than 6 characters and password does not match.

Input-

```
public class w25 {
    public static void main(String[] args) {
        String username = "user"; // Example username
        String password = "pass123";
        String confirmPassword = "pass124"; // Example password

        try {
            if (username.length() < 6) {
                throw new InvalidUsernameException("Username is too short.");
            }
            if (!password.equals(confirmPassword)) {
                throw new PasswordMismatchException("Passwords do not match.");
            }
        } catch (InvalidUsernameException | PasswordMismatchException e) {
            System.out.println(e.getMessage());
        }
    }
}

class InvalidUsernameException extends Exception {
    public InvalidUsernameException(String message) {
        super(message);
    }
}

class PasswordMismatchException extends Exception {
    public PasswordMismatchException(String message) {
        super(message);
    }
}
```

Output-



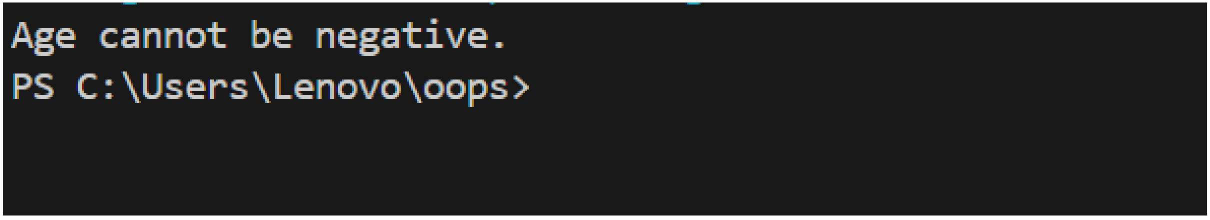
```
Username is too short.
PS C:\Users\Lenovo\oops>
```

26. Write a program to input name and age of a person and throw a user-defined exception, if the entered age is negative

Input-

```
public class w26 {
    public static void main(String[] args) {
        String name = "John";
        int age = -5;

        try {
            if (age < 0) {
                throw new NegativeAgeException("Age cannot be negative.");
            } else {
                System.out.println("Name: " + name + ", Age: " + age);
            }
        } catch (NegativeAgeException e) {
            System.out.println(e.getMessage());
        }
    }

}
class NegativeAgeException extends Exception {
    public NegativeAgeException(String message) {
        super(message);
    }
}
```

Output-



```
Age cannot be negative.
PS C:\Users\Lenovo\oops>
```