Lab Assignments

Name – Jignesh Ameta

Roll No. – 14

Class – MCA 2nd Sem

Subject – JAVA Programming Lab

Date of Submission -

Index

Index

Assignment_01

Pre Required Module – ModuleUtils.java

Code :-

```
package Assignment_01.Modules;
import java.util.Arrays;
public class ModuleUtils {
    // Q1: Sum of N natural numbers
    public static int sumIterative(int n) {
        int sum = 0;
        for (int i = 1; i <= n; i++)
            sum += i;
        return sum;
    }
    public static int sumRecursive(int n) {
        if (n <= 0)
            return 0;
        return n + sumRecursive(n - 1);
    }
    // Q2: Count frequency of positive, negative, zero
    public static int[] countNumbers(int[] arr) {
        int pos = 0, neg = 0, zero = 0;
        for (int num : arr) {
            if (num > 0)
                pos++;
            else if (num < 0)
                neg++;
            else
                zero++;
        }
        return new int[] { pos, neg, zero };
    }
    // Q3: Reverse a number
    public static int reverseIterative(int n) {
        int rev = 0;
        while (n != 0) {
            rev = rev * 10 + n % 10;
            n /= 10;
        }
        return rev;
    }
    public static int reverseRecursive(int n) {
        return reverseRecursiveHelper(n, 0);
    }
    private static int reverseRecursiveHelper(int n, int rev) {
        if (n == 0)
            return rev;
        return reverseRecursiveHelper(n / 10, rev * 10 + n % 10);
    }
    // Q4: Sum of digits
    public static int sumDigitsIterative(int n) {
        int sum = 0;
        n = Math.abs(n);
        while (n != 0) {
            sum += n % 10;
        n /= 10;
    }
    return sum;
}
public static int sumDigitsRecursive(int n) {
    n = Math.abs(n);
    if (n == 0)
        return 0;
    return n % 10 + sumDigitsRecursive(n / 10);
}
// Q5: Even or odd from list
public static int[] evenNumbers(int[] arr) {
    return Arrays.stream(arr).filter(x -> x % 2 == 0).toArray();
}
public static int[] oddNumbers(int[] arr) {
    return Arrays.stream(arr).filter(x -> x % 2 != 0).toArray();
}
// Q6: Factorial & palindrome
public static long factorialIterative(int n) {
    if (n < 0)
        throw new IllegalArgumentException("Negative numbers not allowed");
    long fact = 1;
    for (int i = 2; i <= n; i++)
        fact *= i;
    return fact;
}
public static long factorialRecursive(int n) {
    if (n < 0)
        throw new IllegalArgumentException("Negative numbers not allowed");
    if (n <= 1)
        return 1;
    return n * factorialRecursive(n - 1);
}
public static boolean isPalindrome(int n) {
    n = Math.abs(n);
    return n == reverseIterative(n);
}
// Q7: Patterns
/**
 * Pattern 1: Left-aligned pyramid with spaces and stars (e.g. for n=5)
 *     *
 *    ***
 *   *****
 *  *******
 * *********
 */
public static void pattern1(int n) {
    for (int i = 1; i <= n; i++) {
        // leading spaces
        for (int j = 1; j <= n - i; j++)
            System.out.print(" ");
        // print (2*i - 1) stars
        for (int k = 1; k <= 2 * i - 1; k++)
            System.out.print("*");
        System.out.println();
    }
}
```

```
/**
 * Pattern 2: Left-aligned triangle of stars (e.g. for n=5)
 * *
 * **
 * ***
 * ****
 * *****
 */
public static void pattern2(int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++)
            System.out.print("*");
        System.out.println(" ");
    }
}
/**
 * Pattern 3: Right-aligned triangle of stars (e.g. for n=5)
 *     *
 *    **
 *   ***
 *  ****
 * *****
 */
public static void pattern3(int n) {
    for (int i = 1; i <= n; i++) {
        // leading spaces to push stars to right
        for (int j = 1; j <= n - i; j++)
            System.out.print(" ");
        for (int j = 1; j <= i; j++)
            System.out.print("*");
        System.out.println();
    }
}
// Q8: Max or min from list
public static int max(int[] arr) {
    if (arr == null || arr.length == 0)
        throw new IllegalArgumentException("Array is empty");
    int max = arr[0];
    for (int n : arr)
        if (n > max)
            max = n;
    return max;
}
public static int min(int[] arr) {
    if (arr == null || arr.length == 0)
        throw new IllegalArgumentException("Array is empty");
    int min = arr[0];
    for (int n : arr)
        if (n < min)
            min = n;
    return min;
}
// Q9: Max of three
public static int maxOfThree(int a, int b, int c) {
    return Math.max(a, Math.max(b, c));
}
```

```
  // Q10: Frequency of element
    public static int frequency(int[] arr, int x) {
        int count = 0;
        for (int n : arr)
            if (n == x)
                count++;
        return count;
    }
}
```

Q.1 Sum of N natural numbers ?

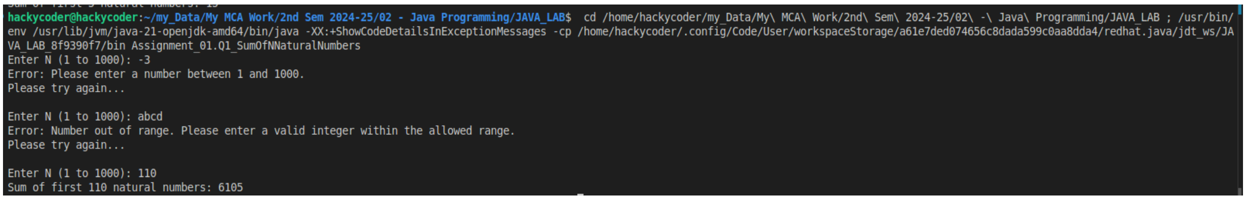
Ans :-

Code :-

```
package Assignment_01;
import java.util.Scanner;
public class Q1_SumOfNNaturalNumbers {
    // Recursive function to calculate sum
    static int sum(int n) {
        if (n <= 1)
            return n;
        return n + sum(n - 1);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = -1;
        while (true) {
            try {
                System.out.print("Enter N (1 to 1000): ");
                String nStr = sc.next();
                try {
                    n = Integer.parseInt(nStr);
                } catch (NumberFormatException ex) {
                    System.out.println("Error: Number out of range. Please enter a valid integer within the allowed
range.");
                    System.out.println("Please try again...\n");
                    continue;
                }
                if (n < 1 || n > 1000) {
                    System.out.println("Error: Please enter a number between 1 and 1000.");
                    System.out.println("Please try again...\n");
                    continue;
                }
                System.out.println("Sum of first " + n + " natural numbers: " + sum(n));
                break; // Exit loop after successful input and output

            } catch (Exception e) {
                System.out.println("Error: Invalid input. Please enter a valid integer.");
                sc.nextLine(); // Clear invalid input from buffer
            }
        }
        sc.close();
    }
}
```

Output :-



```
Sum of first 5 natural numbers: 15
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/
env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JA
VA_LAB_8f9390f7/bin Assignment_01.Q1_SumOfNNaturalNumbers
Enter N (1 to 1000): -3
Error: Please enter a number between 1 and 1000.
Please try again...

Enter N (1 to 1000): abcd
Error: Number out of range. Please enter a valid integer within the allowed range.
Please try again...

Enter N (1 to 1000): 110
Sum of first 110 natural numbers: 6105
```

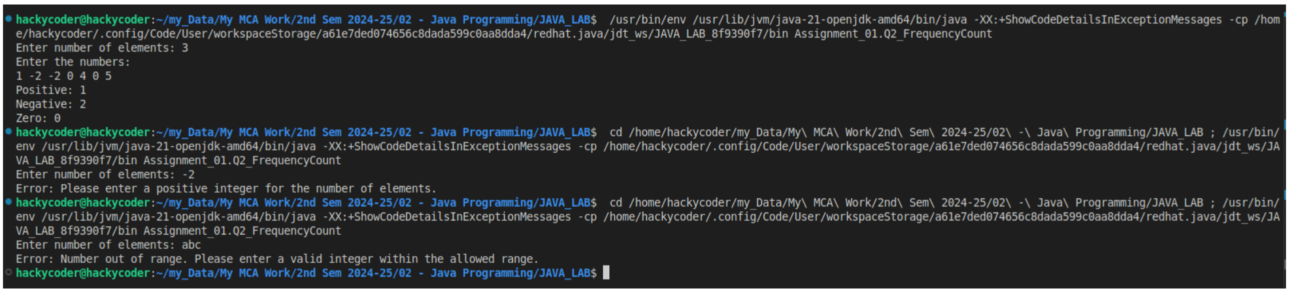
Q.2 Count frequency of positive, negative and zero numbers in a given list of numbers?

Ans :-

Code :-

```
package Assignment_01;
import java.util.Scanner;
public class Q2_FrequencyCount {
    static void countFrequencies(int[] arr) {
        int pos = 0, neg = 0, zero = 0;
        for (int num : arr) {
            if (num > 0) pos++;
            else if (num < 0) neg++;
            else zero++;
        }
        System.out.println("Positive: " + pos);
        System.out.println("Negative: " + neg);
        System.out.println("Zero: " + zero);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter number of elements: ");
            String nStr = sc.next();
            int n = 0;
            int MAX_LIMIT = 1000;
            try {
                n = Integer.parseInt(nStr);
            } catch (NumberFormatException ex) {
                System.out.println("Error: Number out of range. Please enter a valid integer within the allowed range.");
                return;
            }
            if (n <= 0) {
                System.out.println("Error: Please enter a positive integer for the number of elements.");
            } else if (n > MAX_LIMIT) {
                System.out.println("Error: Number of elements cannot exceed " + MAX_LIMIT + ".");
            } else {
                int[] arr = new int[n];
                System.out.println("Enter the numbers:");
                for (int i = 0; i < n; i++) {
                    arr[i] = sc.nextInt();
                }
                countFrequencies(arr);
            }
        } catch (Exception e) {
            System.out.println("Error: Invalid input. Please enter valid integers.");
        } finally {
            sc.close();
        }
    }
}
```

Output :-



```
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /hom
  e/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_01.Q2_FrequencyCount
  Enter number of elements: 3
  Enter the numbers:
  1 -2 -2 0 4 0 5
  Positive: 1
  Negative: 2
  Zero: 0
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/
  env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JA
  VA_LAB_8f9390f7/bin Assignment_01.Q2_FrequencyCount
  Enter number of elements: -2
  Error: Please enter a positive integer for the number of elements.
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/
  env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JA
  VA_LAB_8f9390f7/bin Assignment_01.Q2_FrequencyCount
  Enter number of elements: abc
  Error: Number out of range. Please enter a valid integer within the allowed range.
○ hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ 
```

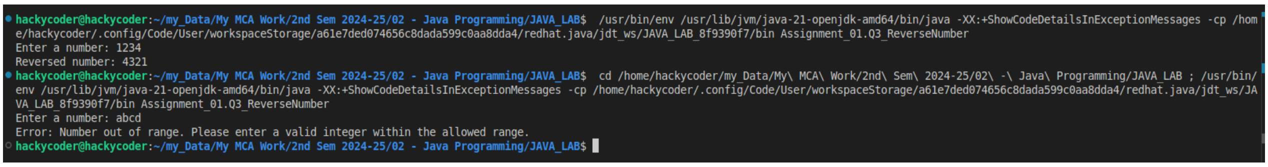
Q.3 Reverse a number ?

Ans :-

Code :-

```
package Assignment_01;
import java.util.Scanner;
public class Q3_ReverseNumber {
    static int reverse(int n) {
        int rev = 0;
        while (n != 0) {
            rev = rev * 10 + n % 10;
            n /= 10;
        }
        return rev;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter a number: ");
            String nStr = sc.next();
            int n = 0;
            try {
                n = Integer.parseInt(nStr);
            } catch (NumberFormatException ex) {
                System.out.println("Error: Number out of range. Please enter a valid integer within the allowed
range.");
                return;
            }
            System.out.println("Reversed number: " + reverse(n));
        } catch (Exception e) {
            System.out.println("Error: Invalid input. Please enter a valid integer.");
        } finally {
            sc.close();
        }
    }
}
```

Output :-



```
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /hom
  e/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_01.Q3_ReverseNumber
  Enter a number: 1234
  Reversed number: 4321
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/
  env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JA
  VA_LAB_8f9390f7/bin Assignment_01.Q3_ReverseNumber
  Enter a number: abcd
  Error: Number out of range. Please enter a valid integer within the allowed range.
○ hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ █
```

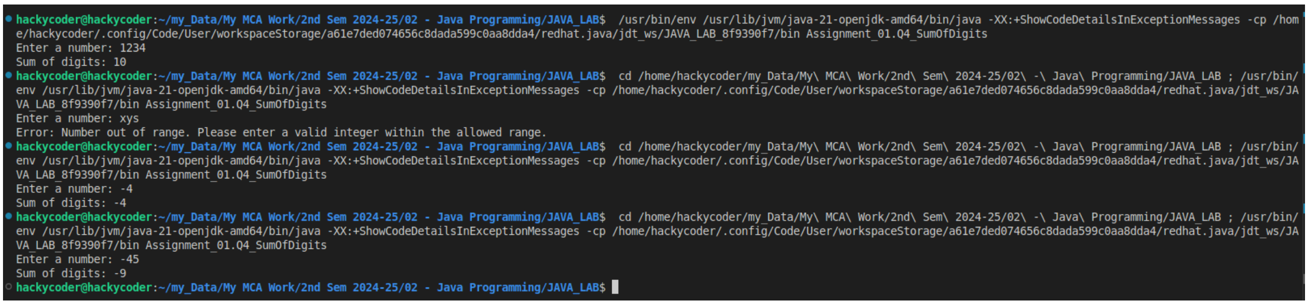
Q.4 sum of digits of a number ?

Ans :-

Code :-

```
package Assignment_01;
import java.util.Scanner;
public class Q4_SumOfDigits {
    // Recursive function to calculate sum of digits
    static int sumDigits(int n) {
        if (n == 0) return 0;
        return n % 10 + sumDigits(n / 10);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter a number: ");
            String nStr = sc.next();
            int n = 0;
            try {
                n = Integer.parseInt(nStr);
            } catch (NumberFormatException ex) {
                System.out.println("Error: Number out of range. Please enter a valid integer within the allowed
range.");
                return;
            }
            System.out.println("Sum of digits: " + sumDigits(n));
        } catch (Exception e) {
            System.out.println("Error: Invalid input. Please enter a valid integer.");
        } finally {
            sc.close();
        }
    }
}
```

Output :-



```
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /hom
  e/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_01.Q4_SumOfDigits
  Enter a number: 1234
  Sum of digits: 10
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/
  env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JA
  VA_LAB_8f9390f7/bin Assignment_01.Q4_SumOfDigits
  Enter a number: xys
  Error: Number out of range. Please enter a valid integer within the allowed range.
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/
  env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JA
  VA_LAB_8f9390f7/bin Assignment_01.Q4_SumOfDigits
  Enter a number: -4
  Sum of digits: -4
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/
  env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JA
  VA_LAB_8f9390f7/bin Assignment_01.Q4_SumOfDigits
  Enter a number: -45
  Sum of digits: -9
○ hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ █
```

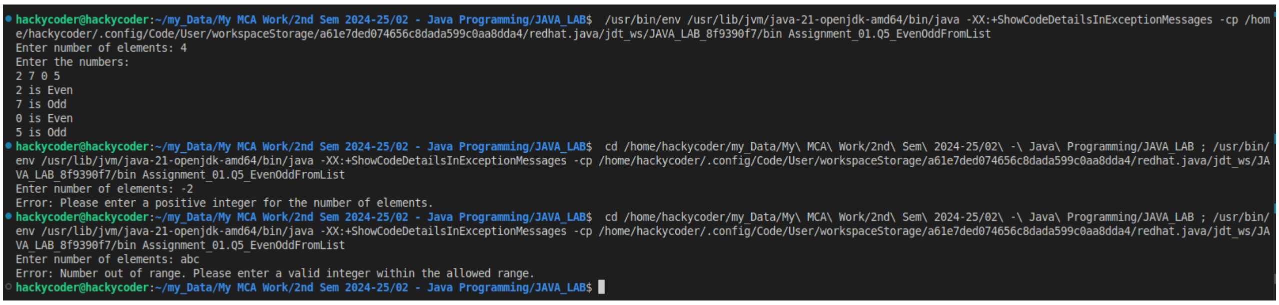
Q.5 even or odd from the list ?

Ans :-

Code :-

```
package Assignment_01;
import java.util.Scanner;
public class Q5_EvenOddFromList {
    static void printEvenOdd(int[] arr) {
        for (int num : arr) {
            if (num % 2 == 0)
                System.out.println(num + " is Even");
            else
                System.out.println(num + " is Odd");
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter number of elements: ");
            String nStr = sc.next();
            int n = 0;
            int MAX_LIMIT = 1000;
            try {
                n = Integer.parseInt(nStr);
            } catch (NumberFormatException ex) {
                System.out.println("Error: Number out of range. Please enter a valid integer within the allowed
range.");
                return;
            }
            if (n <= 0) {
                System.out.println("Error: Please enter a positive integer for the number of elements.");
            } else if (n > MAX_LIMIT) {
                System.out.println("Error: Number of elements cannot exceed " + MAX_LIMIT + ".");
            } else {
                int[] arr = new int[n];
                System.out.println("Enter the numbers:");
                for (int i = 0; i < n; i++) {
                    String numStr = sc.next();
                    try {
                        arr[i] = Integer.parseInt(numStr);
                    } catch (NumberFormatException ex) {
                        System.out.println("Error: Number out of range. Please enter valid integers within the allowed
range.");
                        return;
                    }
                }
                printEvenOdd(arr);
            }
        } catch (Exception e) {
            System.out.println("Error: Invalid input. Please enter valid integers.");
        } finally {
            sc.close();
        }
    }
}
```

Output :-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /hom
e/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_01.Q5_EvenOddFromList
Enter number of elements: 4
Enter the numbers:
2 7 0 5
2 is Even
7 is Odd
0 is Even
5 is Odd
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/
env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JA
VA_LAB_8f9390f7/bin Assignment_01.Q5_EvenOddFromList
Enter number of elements: -2
Error: Please enter a positive integer for the number of elements.
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/
env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JA
VA_LAB_8f9390f7/bin Assignment_01.Q5_EvenOddFromList
Enter number of elements: abc
Error: Number out of range. Please enter a valid integer within the allowed range.
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ 
```

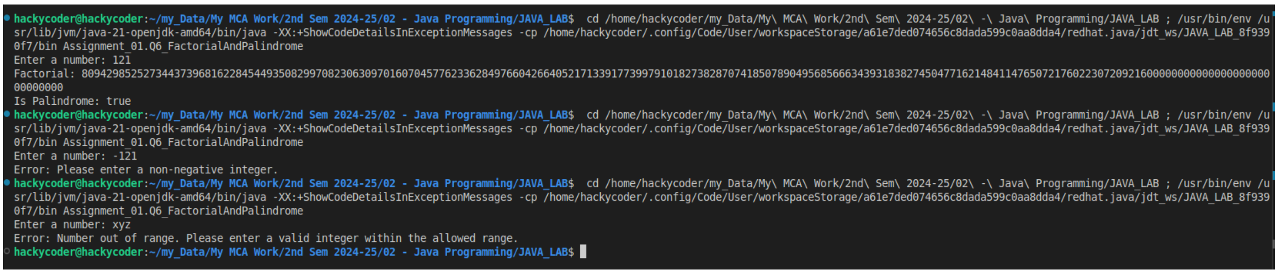
Q.6 find factorial & check palindrome numbers?

Ans :-

Code :-

```
package Assignment_01;
import java.util.Scanner;
public class Q6_FactorialAndPalindrome {
    // Recursive factorial
    static java.math.BigInteger factorial(int n) {
        if (n <= 1) return java.math.BigInteger.ONE;
        return java.math.BigInteger.valueOf(n).multiply(factorial(n - 1));
    }
    static boolean isPalindrome(int n) {
        int original = n, rev = 0;
        while (n != 0) {
            rev = rev * 10 + n % 10;
            n /= 10;
        }
        return original == rev;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter a number: ");
            String nStr = sc.next();
            int n = 0;
            try {
                n = Integer.parseInt(nStr);
            } catch (NumberFormatException ex) {
                System.out.println("Error: Number out of range. Please enter a valid integer within the allowed range.");
                return;
            }
            if (n < 0) {
                System.out.println("Error: Please enter a non-negative integer.");
            } else {
                System.out.println("Factorial: " + factorial(n));
                System.out.println("Is Palindrome: " + isPalindrome(n));
            }
        } catch (Exception e) {
            System.out.println("Error: Invalid input. Please enter a valid integer.");
        } finally {
            sc.close();
        }
    }
}
```

Output :-



```
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/env /u
sr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f939
0f7/bin Assignment_01.Q6_FactorialAndPalindrome
Enter a number: 121
Factorial: 8094298525273443739681622845449350829970823063097016070457762336284976604266405217133917739979101827382870741850789049568566634393183827450477162148411476507217602230720921600000000000000000000
00000000
Is Palindrome: true
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/env /u
sr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f939
0f7/bin Assignment_01.Q6_FactorialAndPalindrome
Enter a number: -121
Error: Please enter a non-negative integer.
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/env /u
sr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f939
0f7/bin Assignment_01.Q6_FactorialAndPalindrome
Enter a number: xyz
Error: Number out of range. Please enter a valid integer within the allowed range.
○ hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ ▌
```

Q.7 Java Program to draw the following patterns

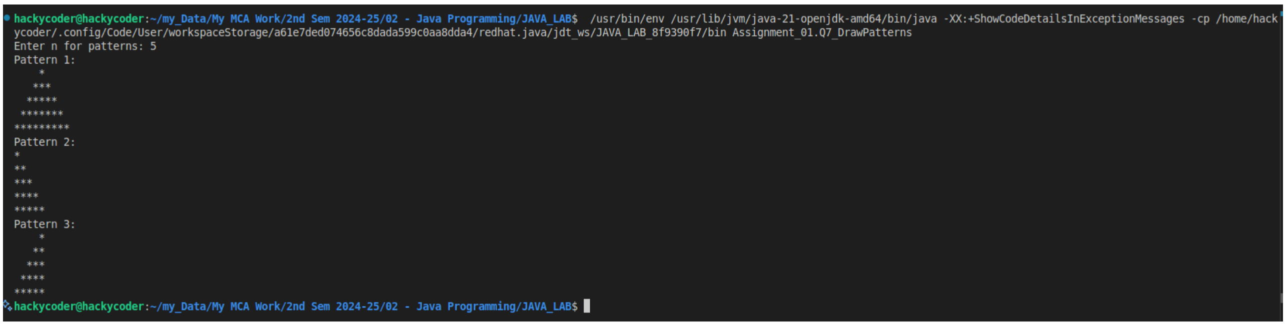
```
    *         *            *
   * *        **          **
  * * *       ***        ***
 * * * *      ****      ****
* * * * *     *****    *****
```

Ans :-

Code :-

```
package Assignment_01;
import java.util.Scanner;
public class Q7_DrawPatterns {
    static void pattern1(int n) {
           for (int i = 1; i <= n; i++) {
            // leading spaces
            for (int j = 1; j <= n - i; j++) System.out.print(" ");
            // print (2*i - 1) stars
            for (int k = 1; k <= 2 * i - 1; k++) System.out.print("*");
            System.out.println();
        }
    }
    static void pattern2(int n) {
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= i; j++) System.out.print("*");
            System.out.println(" ");
        }
    }
    static void pattern3(int n) {
        for (int i = 1; i <= n; i++) {
            // leading spaces to push stars to right
            for (int j = 1; j <= n - i; j++) System.out.print(" ");
            for (int j = 1; j <= i; j++) System.out.print("*");
            System.out.println();
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter n for patterns: ");
            String nStr = sc.next();
            int n = 0;
            int MAX_LIMIT = 1000;
            try {
                n = Integer.parseInt(nStr);
            } catch (NumberFormatException ex) {
                System.out.println("Error: Number out of range. Please enter a valid integer within the allowed
range.");
                return;
            }
            if (n <= 0) {
                System.out.println("Error: Please enter a positive integer for n.");
            } else if (n > MAX_LIMIT) {
                System.out.println("Error: n cannot exceed " + MAX_LIMIT + ".");
            } else {
                System.out.println("Pattern 1:");
                pattern1(n);
                System.out.println("Pattern 2:");
                pattern2(n);
                System.out.println("Pattern 3:");
                pattern3(n);
            }
        } catch (Exception e) {
            System.out.println("Error: Invalid input. Please enter a valid integer.");
            sc.next(); // Clear the invalid input
        } finally {
            sc.close();
        }
    }
}
```

Output :-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hack
ycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_01.Q7_DrawPatterns
Enter n for patterns: 5
Pattern 1:
    *
   ***
  *****
 *******
*********
Pattern 2:
*
**
***
****
*****
Pattern 3:
    *
   **
  ***
 ****
*****
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ 
```

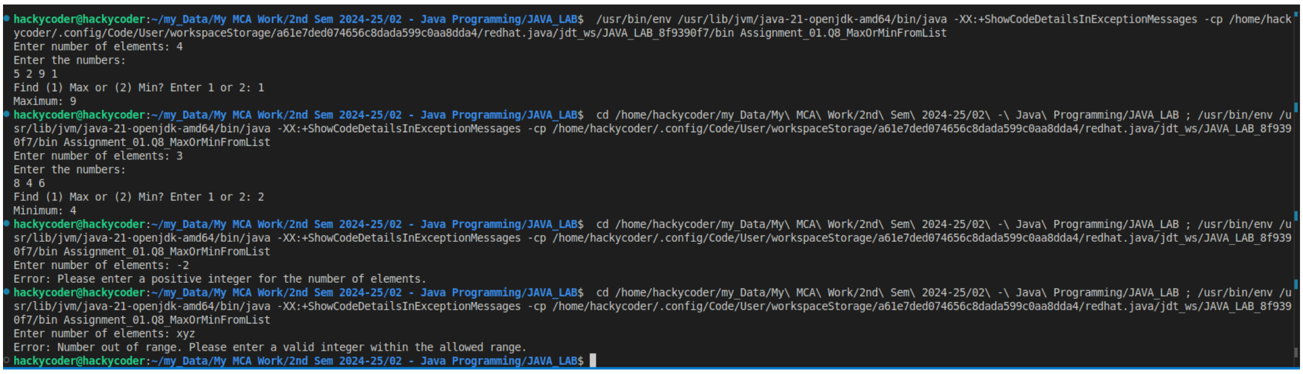
Q.8 max or min from the list ?

Ans :-

Code :-

```
package Assignment_01;
import java.util.Scanner;
public class Q8_MaxOrMinFromList {
    static int findMax(int[] arr) {
        int max = arr[0];
        for (int num : arr) if (num > max) max = num;
        return max;
    }
    static int findMin(int[] arr) {
        int min = arr[0];
        for (int num : arr) if (num < min) min = num;
        return min;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter number of elements: ");
            String nStr = sc.next();
            int n = 0;
            int MAX_LIMIT = 1000;
            try {
                n = Integer.parseInt(nStr);
            } catch (NumberFormatException ex) {
                System.out.println("Error: Number out of range. Please enter a valid integer within the allowed
range.");
                return;
            }
            if (n <= 0) {
                System.out.println("Error: Please enter a positive integer for the number of elements.");
            } else if (n > MAX_LIMIT) {
                System.out.println("Error: Number of elements cannot exceed " + MAX_LIMIT + ".");
            } else {
                int[] arr = new int[n];
                System.out.println("Enter the numbers:");
                for (int i = 0; i < n; i++) {
                    String numStr = sc.next();
                    try {
                        arr[i] = Integer.parseInt(numStr);
                    } catch (NumberFormatException ex) {
                        System.out.println("Error: Number out of range. Please enter valid integers within the allowed
range.");
                        return;
                    }
                }
                System.out.print("Find (1) Max or (2) Min? Enter 1 or 2: ");
                String choiceStr = sc.next();
                int choice;
                try {
                    choice = Integer.parseInt(choiceStr);
                } catch (NumberFormatException ex) {
                    System.out.println("Error: Number out of range. Please enter 1 or 2.");
                    return;
                }
                if (choice == 1)
                    System.out.println("Maximum: " + findMax(arr));
                else if (choice == 2)
                    System.out.println("Minimum: " + findMin(arr));
                else
                    System.out.println("Error: Invalid choice. Enter 1 for Max or 2 for Min.");
            }
        } catch (Exception e) {
            System.out.println("Error: Invalid input. Please enter valid integers.");
        } finally {
            sc.close();
        }
    }
}
```

Output :-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hack
ycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_01.Q8_MaxOrMinFromList
Enter number of elements: 4
Enter the numbers:
5 2 9 1
Find (1) Max or (2) Min? Enter 1 or 2: 1
Maximum: 9
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/env /u
sr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f939
0f7/bin Assignment_01.Q8_MaxOrMinFromList
Enter number of elements: 3
Enter the numbers:
8 4 6
Find (1) Max or (2) Min? Enter 1 or 2: 2
Minimum: 4
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/env /u
sr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f939
0f7/bin Assignment_01.Q8_MaxOrMinFromList
Enter number of elements: -2
Error: Please enter a positive integer for the number of elements.
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/env /u
sr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f939
0f7/bin Assignment_01.Q8_MaxOrMinFromList
Enter number of elements: xyz
Error: Number out of range. Please enter a valid integer within the allowed range.
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ 
```

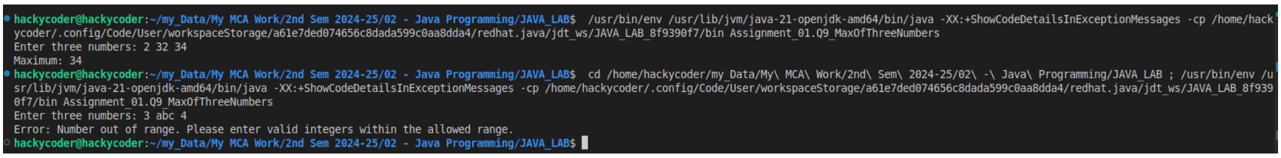
Q.9 find maximum of the three numbers ?

Ans :-

Code :-

```
package Assignment_01;
import java.util.Scanner;
public class Q9_MaxOfThreeNumbers {
    static int maxOfThree(int a, int b, int c) {
        return Math.max(a, Math.max(b, c));
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter three numbers: ");
            String aStr = sc.next();
            String bStr = sc.next();
            String cStr = sc.next();
            int a, b, c;
            try {
                a = Integer.parseInt(aStr);
                b = Integer.parseInt(bStr);
                c = Integer.parseInt(cStr);
            } catch (NumberFormatException ex) {
                System.out.println("Error: Number out of range. Please enter valid integers within the allowed range.");
                return;
            }
            System.out.println("Maximum: " + maxOfThree(a, b, c));
        } catch (Exception e) {
            System.out.println("Error: Invalid input. Please enter three valid integers.");
        } finally {
            sc.close();
        }
    }
}
```

Output :-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hack
ycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_01.Q9_MaxOfThreeNumbers
Enter three numbers: 2 32 34
Maximum: 34
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/env /u
sr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f939
0f7/bin Assignment_01.Q9_MaxOfThreeNumbers
Enter three numbers: 3 abc 4
Error: Number out of range. Please enter valid integers within the allowed range.
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ 
```

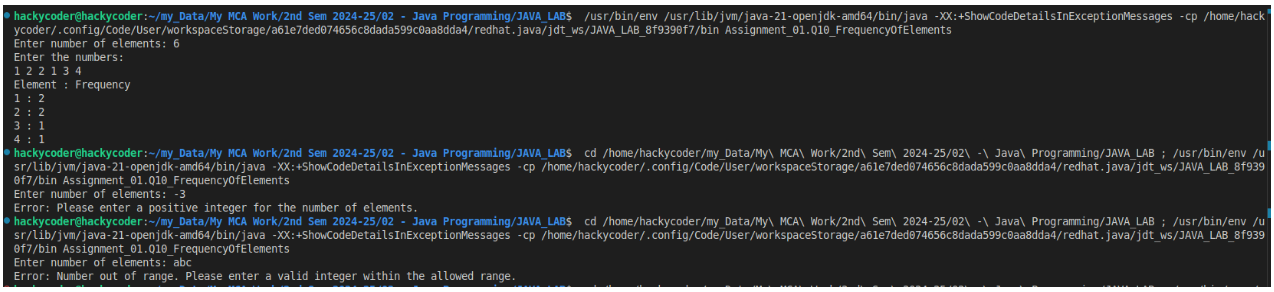
Q.10 count the frequency of the element in the list ?

Ans :-

Code :-

```
package Assignment_01;
import java.util.HashMap;
import java.util.Scanner;
public class Q10_FrequencyOfElements {
    static void countFrequency(int[] arr) {
        HashMap<Integer, Integer> freq = new HashMap<>();
        for (int num : arr) {
            freq.put(num, freq.getOrDefault(num, 0) + 1);
        }
        System.out.println("Element : Frequency");
        for (int key : freq.keySet()) {
            System.out.println(key + " : " + freq.get(key));
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter number of elements: ");
            String nStr = sc.next();
            int n = 0;
            int MAX_LIMIT = 1000;
            try {
                n = Integer.parseInt(nStr);
            } catch (NumberFormatException ex) {
                System.out.println("Error: Number out of range. Please enter a valid integer within the allowed range.");
                return;
            }
            if (n <= 0) {
                System.out.println("Error: Please enter a positive integer for the number of elements.");
            } else if (n > MAX_LIMIT) {
                System.out.println("Error: Number of elements cannot exceed " + MAX_LIMIT + ".");
            } else {
                int[] arr = new int[n];
                System.out.println("Enter the numbers:");
                for (int i = 0; i < n; i++) {
                    String numStr = sc.next();
                    try {
                        arr[i] = Integer.parseInt(numStr);
                    } catch (NumberFormatException ex) {
                        System.out.println("Error: Number out of range. Please enter valid integers within the allowed range.");
                        return;
                    }
                }
                countFrequency(arr);
            }
        } catch (Exception e) {
            System.out.println("Error: Invalid input. Please enter valid integers.");
        } finally {
            sc.close();
        }
    }
}
```

Output :-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hack
ycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_01.Q10_FrequencyOfElements
Enter number of elements: 6
Enter the numbers:
1 2 2 1 3 4
Element : Frequency
1 : 2
2 : 2
3 : 1
4 : 1
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/env /u
sr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f939
0f7/bin Assignment_01.Q10_FrequencyOfElements
Enter number of elements: -3
Error: Please enter a positive integer for the number of elements.
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/env /u
sr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f939
0f7/bin Assignment_01.Q10_FrequencyOfElements
Enter number of elements: abc
Error: Number out of range. Please enter a valid integer within the allowed range.
```

Q.11 Selection Option Based Modularized Program ?

Ans :-

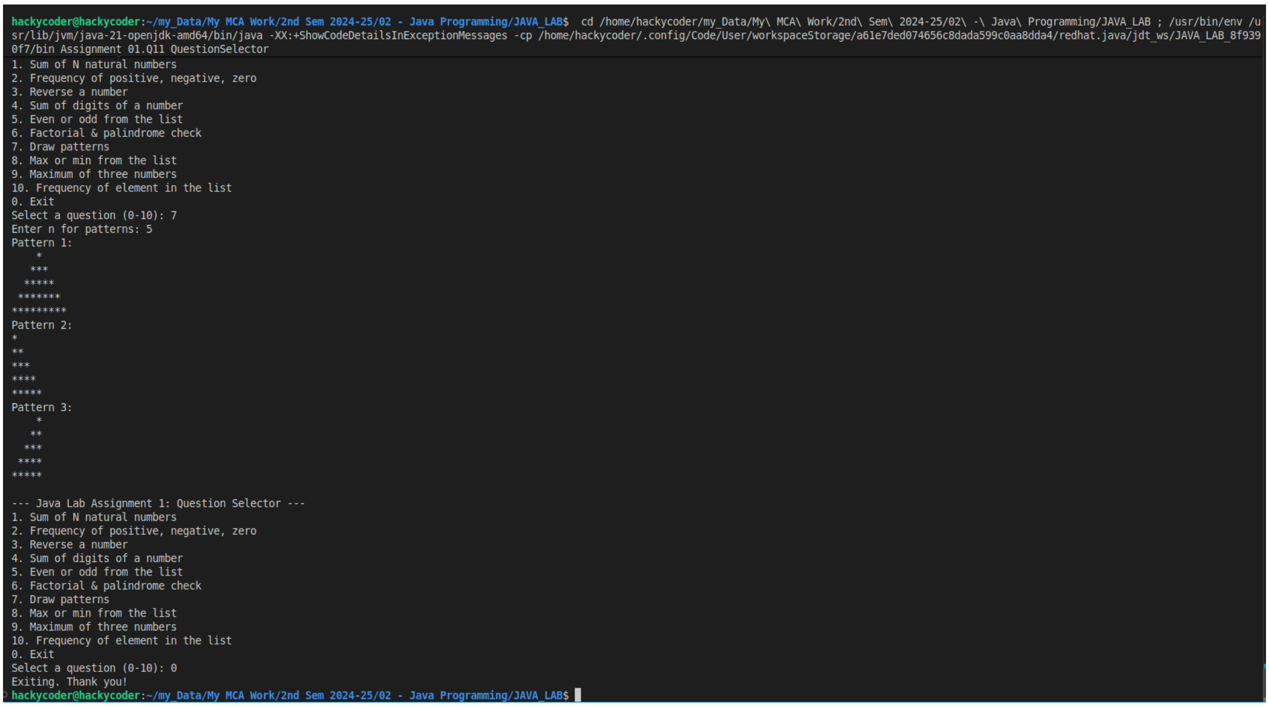
Code :-

```
package Assignment_01;
import java.util.Scanner;
import java.util.Arrays;
import Assignment_01.Modules.ModuleUtils;
public class Q11_QuestionSelector {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        while (true) {
            try {
                System.out.println("\n--- Java Lab Assignment 1: Question Selector ---");
                System.out.println("1. Sum of N natural numbers");
                System.out.println("2. Frequency of positive, negative, zero");
                System.out.println("3. Reverse a number");
                System.out.println("4. Sum of digits of a number");
                System.out.println("5. Even or odd from the list");
                System.out.println("6. Factorial & palindrome check");
                System.out.println("7. Draw patterns");
                System.out.println("8. Max or min from the list");
                System.out.println("9. Maximum of three numbers");
                System.out.println("10. Frequency of element in the list");
                System.out.println("0. Exit");
                System.out.print("Select a question (0-10): ");
                int choice = sc.nextInt();
                if (choice == 0) {
                    System.out.println("Exiting. Thank you!");
                    break;
                }
                switch (choice) {
                    case 1:
                        System.out.print("Enter N: ");
                        int n = sc.nextInt();
                        if (n <= 0) {
                            System.out.println("Error: Please enter a positive integer greater than 0.");
                        } else {
                            System.out.println("Sum (iterative): " + ModuleUtils.sumIterative(n));
                            System.out.println("Sum (recursive): " + ModuleUtils.sumRecursive(n));
                        }
                        break;
                    case 2:
```

```
    System.out.print("Enter number of elements: ");
    int n2 = sc.nextInt();
    if (n2 <= 0) {
        System.out.println("Error: Please enter a positive integer for the number of elements.");
    } else {
        int[] arr2 = new int[n2];
        System.out.println("Enter the numbers:");
        for (int i = 0; i < n2; i++)
            arr2[i] = sc.nextInt();
        int[] freq = ModuleUtils.countNumbers(arr2);
        System.out.println("Positive: " + freq[0] + ", Negative: " + freq[1] + ", Zero: " + freq[2]);
    }
    break;
case 3:
    System.out.print("Enter a number: ");
    int n3 = sc.nextInt();
    System.out.println("Reverse (iterative): " + ModuleUtils.reverseIterative(n3));
    System.out.println("Reverse (recursive): " + ModuleUtils.reverseRecursive(n3));
    break;
case 4:
    System.out.print("Enter a number: ");
    int n4 = sc.nextInt();
    System.out.println("Sum of digits (iterative): " + ModuleUtils.sumDigitsIterative(n4));
    System.out.println("Sum of digits (recursive): " + ModuleUtils.sumDigitsRecursive(n4));
    break;
case 5:
    System.out.print("Enter number of elements: ");
    int n5 = sc.nextInt();
    if (n5 <= 0) {
        System.out.println("Error: Please enter a positive integer for the number of elements.");
    } else {
        int[] arr5 = new int[n5];
        System.out.println("Enter the numbers:");
        for (int i = 0; i < n5; i++)
            arr5[i] = sc.nextInt();
        System.out.println("Even numbers: " + Arrays.toString(ModuleUtils.evenNumbers(arr5)));
        System.out.println("Odd numbers: " + Arrays.toString(ModuleUtils.oddNumbers(arr5)));
    }
    break;
case 6:
    System.out.print("Enter a number: ");
    int n6 = sc.nextInt();
    if (n6 < 0) {
        System.out.println("Error: Please enter a non-negative integer.");
    } else {
        System.out.println("Factorial (iterative): " + ModuleUtils.factorialIterative(n6));
        System.out.println("Factorial (recursive): " + ModuleUtils.factorialRecursive(n6));
        System.out.println("Is Palindrome: " + ModuleUtils.isPalindrome(n6));
    }
    break;
case 7:
    System.out.print("Enter n for patterns: ");
    int n7 = sc.nextInt();
    if (n7 <= 0) {
        System.out.println("Error: Please enter a positive integer for n.");
    } else {
        System.out.println("Pattern 1:");
```

```
                            ModuleUtils.pattern1(n7);
                            System.out.println("Pattern 2:");
                            ModuleUtils.pattern2(n7);
                            System.out.println("Pattern 3:");
                            ModuleUtils.pattern3(n7);
                        }
                        break;
                    case 8:
                        System.out.print("Enter number of elements: ");
                        int n8 = sc.nextInt();
                        int[] arr8 = new int[n8];
                        System.out.println("Enter the numbers:");
                        for (int i = 0; i < n8; i++)
                            arr8[i] = sc.nextInt();
                        System.out.print("Find (1) Max or (2) Min? Enter 1 or 2: ");
                        int ch8 = sc.nextInt();
                        if (ch8 == 1)
                            System.out.println("Maximum: " + ModuleUtils.max(arr8));
                        else
                            System.out.println("Minimum: " + ModuleUtils.min(arr8));
                        break;
                    case 9:
                        System.out.print("Enter three numbers: ");
                        int a = sc.nextInt(), b = sc.nextInt(), c = sc.nextInt();
                        System.out.println("Maximum: " + ModuleUtils.maxOfThree(a, b, c));
                        break;
                    case 10:
                        System.out.print("Enter number of elements: ");
                        int n10 = sc.nextInt();
                        int[] arr10 = new int[n10];
                        System.out.println("Enter the numbers:");
                        for (int i = 0; i < n10; i++)
                            arr10[i] = sc.nextInt();
                        System.out.print("Enter element to count frequency: ");
                        int x = sc.nextInt();
                        System.out.println("Frequency of " + x + ": " + ModuleUtils.frequency(arr10, x));
                        break;
                    default:
                        System.out.println("Invalid option. Try again.");
                }
            } catch (Exception e) {
                System.out.println("Error: Invalid input. Please enter valid integers.");
                sc.next(); // Clear the invalid input
            }
        }
        sc.close();
    }
}
```

Output :-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/bin/env /u
sr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f939
0f7/bin Assignment_01.Q11 QuestionSelector
1. Sum of N natural numbers
2. Frequency of positive, negative, zero
3. Reverse a number
4. Sum of digits of a number
5. Even or odd from the list
6. Factorial & palindrome check
7. Draw patterns
8. Max or min from the list
9. Maximum of three numbers
10. Frequency of element in the list
0. Exit
Select a question (0-10): 7
Enter n for patterns: 5
Pattern 1:
    *
   ***
  *****
 *******
*********
Pattern 2:
*
**
***
****
*****
Pattern 3:
    *
   **
  ***
 ****
*****

--- Java Lab Assignment 1: Question Selector ---
1. Sum of N natural numbers
2. Frequency of positive, negative, zero
3. Reverse a number
4. Sum of digits of a number
5. Even or odd from the list
6. Factorial & palindrome check
7. Draw patterns
8. Max or min from the list
9. Maximum of three numbers
10. Frequency of element in the list
0. Exit
Select a question (0-10): 0
Exiting. Thank you!
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ 
```

Assignment_02

Q.1 Write a Java program that reads a line of text from the keyboard and displays the following:

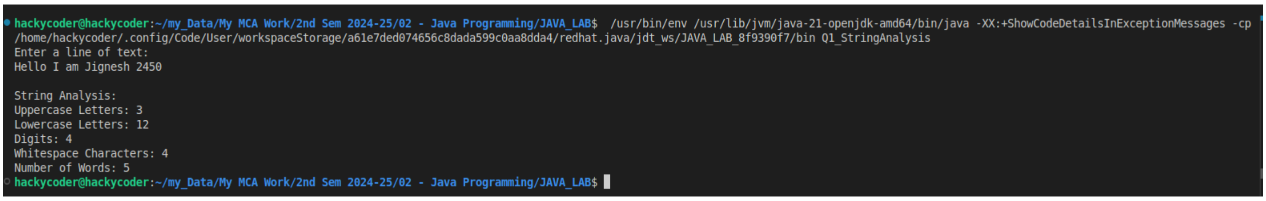
- **Number of uppercase letters**
- **Number of lowercase letters**
- **Number of digits**
- **Number of whitespace characters**
- **Number of words in the string**

Ans:-

Code:-

```
import java.util.*;
public class Q1_StringAnalysis {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a line of text:");
        String text = sc.nextLine();
        int upper = 0, lower = 0, digits = 0, spaces = 0;
        // Loop to count uppercase, lowercase, digits, and spaces
        for (char c : text.toCharArray()) {
            if (Character.isUpperCase(c)) upper++;
            else if (Character.isLowerCase(c)) lower++;
            else if (Character.isDigit(c)) digits++;
            else if (Character.isWhitespace(c)) spaces++;
        }
        sc.close();
        // Count words based on whitespace separation
        int words = text.trim().isEmpty() ? 0 : text.trim().split("\\s+").length;
        // Output the results
        System.out.println("\nString Analysis:");
        System.out.println("Uppercase Letters: " + upper);
        System.out.println("Lowercase Letters: " + lower);
        System.out.println("Digits: " + digits);
        System.out.println("Whitespace Characters: " + spaces);
        System.out.println("Number of Words: " + words);
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp
/home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Q1_StringAnalysis
Enter a line of text:
Hello I am Jignesh 2450

String Analysis:
Uppercase Letters: 3
Lowercase Letters: 12
Digits: 4
Whitespace Characters: 4
Number of Words: 5
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

Q.2 Write a Java program to validate a password entered by the user. The password must satisfy the following conditions:

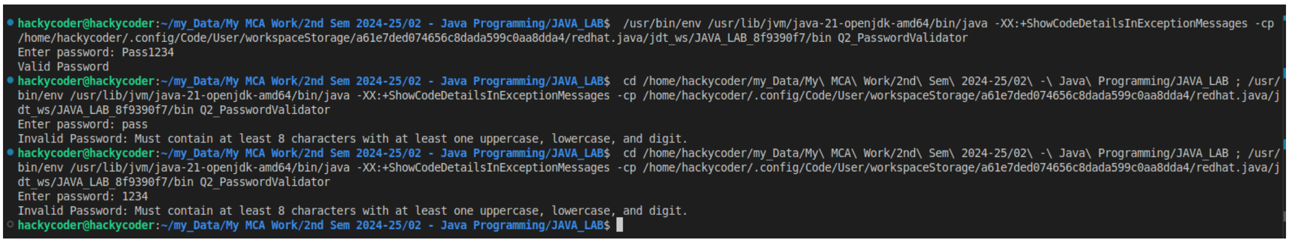
- **Must be at least 8 characters long**
- **Must contain at least one uppercase letter**
- **Must contain at least one lowercase letter**
- **Must contain at least one digit**

Ans:-

Code:-

```
import java.util.*;
class Q2_PasswordValidator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter password: ");
        String password = sc.nextLine();
        boolean hasUpper = false, hasLower = false, hasDigit = false;
        // Check each character in the password
        if (password.length() >= 8) {
            for (char c : password.toCharArray()) {
                if (Character.isUpperCase(c)) hasUpper = true;
                if (Character.isLowerCase(c)) hasLower = true;
                if (Character.isDigit(c)) hasDigit = true;
            }
        }
        sc.close();
        // Validate based on conditions
        if (hasUpper && hasLower && hasDigit && password.length() >= 8) {
            System.out.println("Valid Password");
        } else {
            System.out.println("Invalid Password: Must contain at least 8 characters with at least one uppercase, 
lowercase, and digit.");
        }
    }
}
```

Output:-



```
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp
  /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Q2_PasswordValidator
  Enter password: Pass1234
  Valid Password
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/
  bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/j
  dt_ws/JAVA_LAB_8f9390f7/bin Q2_PasswordValidator
  Enter password: pass
  Invalid Password: Must contain at least 8 characters with at least one uppercase, lowercase, and digit.
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/
  bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/j
  dt_ws/JAVA_LAB_8f9390f7/bin Q2_PasswordValidator
  Enter password: 1234
  Invalid Password: Must contain at least 8 characters with at least one uppercase, lowercase, and digit.
○ hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ 
```

Q.3 Write a Java program to perform the following operations on a list (used in place of a tuple):

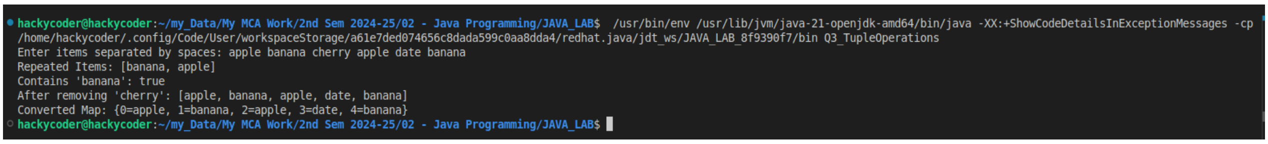
- **Find repeated items**
- **Check whether an element exists**
- **Remove a specific item**
- **Convert the list to a dictionary-like map (index as key, element as value)**

Ans:-

Code:-

```
import java.util.*;
class Q3_TupleOperations {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter items separated by spaces: ");
        String input = sc.nextLine();
        List<String> list = new ArrayList<>(Arrays.asList(input.split("\\s+")));
        // Find repeated items
        Set<String> seen = new HashSet<>();
        Set<String> repeated = new HashSet<>();
        for (String item : list) {
            if (!seen.add(item)) repeated.add(item);
        }
        System.out.println("Repeated Items: " + repeated);
        // Check if an item exists
        String toCheck = "banana";
        System.out.println("Contains '" + toCheck + "': " + list.contains(toCheck));
        // Remove an item
        String toRemove = "cherry";
        list.remove(toRemove);
        System.out.println("After removing '" + toRemove + "': " + list);
        // Convert list to map (index -> value)
        Map<Integer, String> map = new HashMap<>();
        for (int i = 0; i < list.size(); i++) {
            map.put(i, list.get(i));
        }
        System.out.println("Converted Map: " + map);
        sc.close();
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp
/home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Q3_TupleOperations
Enter items separated by spaces: apple banana cherry apple date banana
Repeated Items: [banana, apple]
Contains 'banana': true
After removing 'cherry': [apple, banana, apple, date, banana]
Converted Map: {0=apple, 1=banana, 2=apple, 3=date, 4=banana}
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

Q.4 Write a Java program to implement the Binary Search algorithm on an array of integers. The program should:

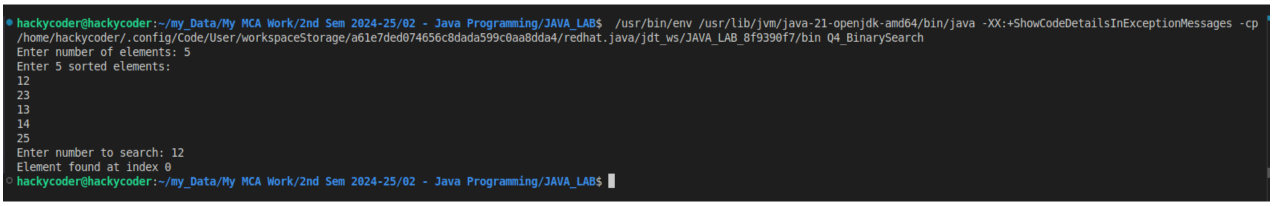
- **Accept a sorted array from the user**
- **Accept a key to be searched**
- **Display whether the key exists or not, and if yes, its position**

Ans:-

Code:-

```
import java.util.*;
class Q4_BinarySearch {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter " + n + " sorted elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.print("Enter number to search: ");
        int key = sc.nextInt();
        int low = 0, high = arr.length - 1;
        boolean found = false;
        // Standard binary search loop
        while (low <= high) {
            int mid = (low + high) / 2;
            if (arr[mid] == key) {
                found = true;
                System.out.println("Element found at index " + mid);
                break;
            } else if (key < arr[mid]) {
                high = mid - 1;
            } else {
                low = mid + 1;
            }
        }
        sc.close();
        if (!found) System.out.println("Element not found");
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Q4_BinarySearch
Enter number of elements: 5
Enter 5 sorted elements:
12
23
13
14
25
Enter number to search: 12
Element found at index 0
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

Q.5 Write a Java program to implement the Linear Search algorithm on an array. The program should:

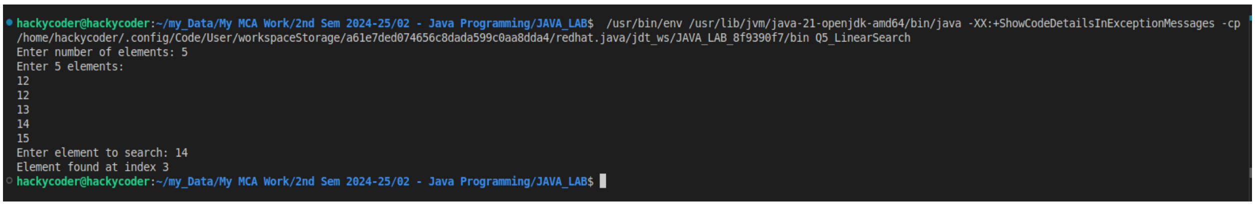
- **Accept an array from the user**
- **Accept a key to be searched**
- **Display the position of the key if found**

Ans:-

Code:-

```
import java.util.*;
class Q5_LinearSearch {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.print("Enter element to search: ");
        int key = sc.nextInt();
        boolean found = false;
        // Traverse and compare each element
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == key) {
                System.out.println("Element found at index " + i);
                found = true;
                break;
            }
        }
        sc.close();
        if (!found) System.out.println("Element not found");
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp
/home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Q5_LinearSearch
Enter number of elements: 5
Enter 5 elements:
12
12
13
14
15
Enter element to search: 14
Element found at index 3
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

Q.6 Write a Java program to perform Selection Sort on an array of integers. The program should:

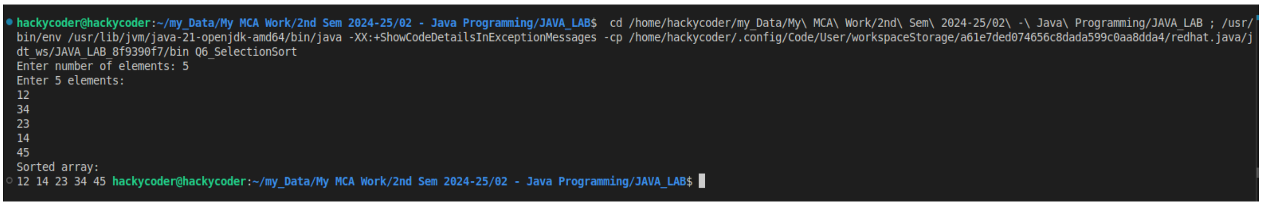
- **Accept input from the user**
- **Display the sorted array**

Ans:-

Code:-

```
class Q6_SelectionSort {
    public static void main(String[] args) {
        java.util.Scanner sc = new java.util.Scanner(System.in);
        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        // Outer loop selects each element
        for (int i = 0; i < n - 1; i++) {
            int minIdx = i;
            // Find the smallest element in unsorted array
            for (int j = i + 1; j < n; j++) {
                if (arr[j] < arr[minIdx]) minIdx = j;
            }
            // Swap
            int temp = arr[minIdx];
            arr[minIdx] = arr[i];
            arr[i] = temp;
        }
        System.out.println("Sorted array:");
        for (int val : arr) System.out.print(val + " ");
        sc.close();
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  cd /home/hackycoder/my_Data/My\ MCA\ Work/2nd\ Sem\ 2024-25/02\ -\ Java\ Programming/JAVA_LAB ; /usr/
bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/j
dt_ws/JAVA_LAB_8f9390f7/bin Q6_SelectionSort
Enter number of elements: 5
Enter 5 elements:
12
34
23
14
45
Sorted array:
12 14 23 34 45 hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ 
```

Q.7 Write a Java program to implement Merge Sort recursively on an array. The program should:

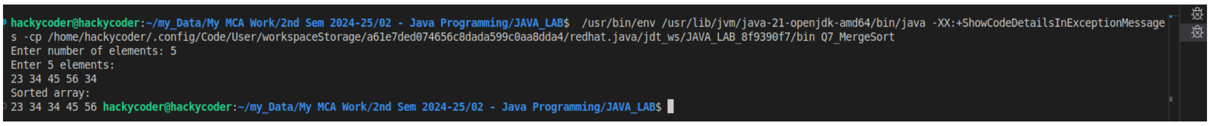
- **Accept input from the user**
- **Display the sorted output**

Ans:-

Code:-

```
class Q7_MergeSort {
    // Recursive merge sort function
    public static void mergeSort(int[] arr, int l, int r) {
        if (l < r) {
            int m = (l + r) / 2;
            mergeSort(arr, l, m);
            mergeSort(arr, m + 1, r);
            merge(arr, l, m, r);
        }
    }
    // Merge two sorted halves
    public static void merge(int[] arr, int l, int m, int r) {
        int n1 = m - l + 1;
        int n2 = r - m;
        int[] L = new int[n1];
        int[] R = new int[n2];
        for (int i = 0; i < n1; i++) L[i] = arr[l + i];
        for (int j = 0; j < n2; j++) R[j] = arr[m + 1 + j];
        int i = 0, j = 0, k = l;
        while (i < n1 && j < n2) {
            if (L[i] <= R[j]) arr[k++] = L[i++];
            else arr[k++] = R[j++];
        }
        while (i < n1) arr[k++] = L[i++];
        while (j < n2) arr[k++] = R[j++];
    }
    public static void main(String[] args) {
        java.util.Scanner sc = new java.util.Scanner(System.in);
        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        mergeSort(arr, 0, arr.length - 1);
        System.out.println("Sorted array:");
        for (int val : arr) System.out.print(val + " ");
        sc.close();
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessage
s -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Q7_MergeSort
Enter number of elements: 5
Enter 5 elements:
23 34 45 56 34
Sorted array:
23 34 34 45 56 hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ 
```

Q.8 Write a Java program to implement Quick Sort recursively on an array. The program should:

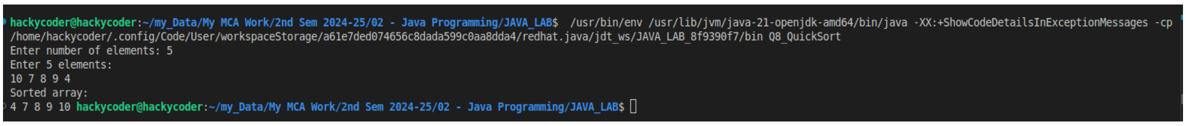
- **Accept input from the user**
- **Display the sorted array after applying Quick Sort**

Ans:-

Code:-

```
class Q8_QuickSort {
    // Recursive quicksort function
    public static void quickSort(int[] arr, int low, int high) {
        if (low < high) {
            int pi = partition(arr, low, high);
            quickSort(arr, low, pi - 1);
            quickSort(arr, pi + 1, high);
        }
    }
    // Partition the array using last element as pivot
    public static int partition(int[] arr, int low, int high) {
        int pivot = arr[high];
        int i = (low - 1);
        for (int j = low; j < high; j++) {
            if (arr[j] <= pivot) {
                i++;
                int temp = arr[i]; arr[i] = arr[j]; arr[j] = temp;
            }
        }
        int temp = arr[i + 1]; arr[i + 1] = arr[high]; arr[high] = temp;
        return i + 1;
    }
    public static void main(String[] args) {
        java.util.Scanner sc = new java.util.Scanner(System.in);
        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        quickSort(arr, 0, arr.length - 1);
        System.out.println("Sorted array:");
        for (int val : arr) System.out.print(val + " ");
        sc.close();
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp 
/home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Q8_QuickSort
Enter number of elements: 5
Enter 5 elements:
10 7 8 9 4
Sorted array:
4 7 8 9 10 hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ 
```

Q.9 Write a Java program to implement Shell Sort on an array. The program should:

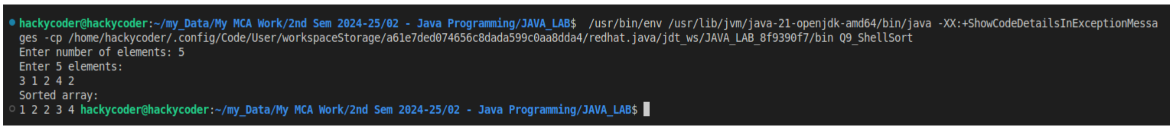
- **Accept input from the user**
- **Display the sorted output after using Shell Sort**

Ans:-

Code:-

```
class Q9_ShellSort {
    public static void main(String[] args) {
        java.util.Scanner sc = new java.util.Scanner(System.in);
        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        // Shell sort with gap reduction
        for (int gap = n / 2; gap > 0; gap /= 2) {
            for (int i = gap; i < n; i++) {
                int temp = arr[i];
                int j;
                for (j = i; j >= gap && arr[j - gap] > temp; j -= gap) {
                    arr[j] = arr[j - gap];
                }
                arr[j] = temp;
            }
        }
        System.out.println("Sorted array:");
        for (int val : arr) System.out.print(val + " ");
        sc.close();
    }
}
```

Output:-



```
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessa
  ges -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Q9_ShellSort
  Enter number of elements: 5
  Enter 5 elements:
  3 1 2 4 2
  Sorted array:
○ 1 2 2 3 4 hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ █
```

Q.10 Write a menu-driven Java program to perform the following matrix operations:

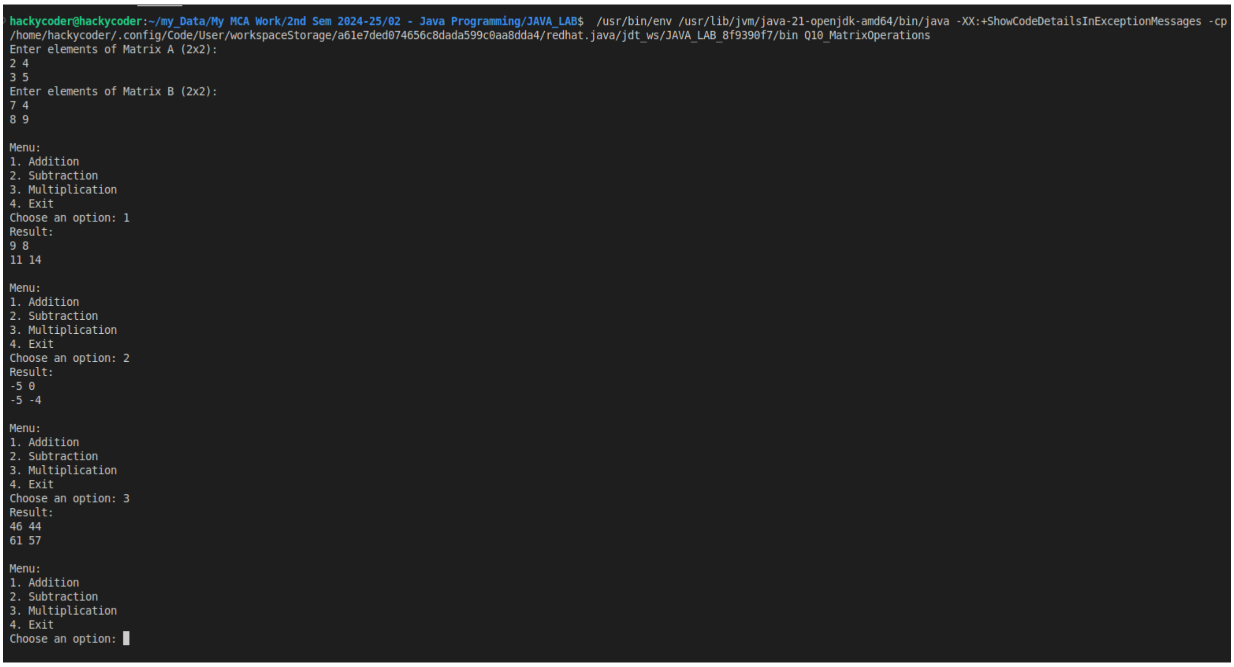
- **Addition of two matrices**
- **Subtraction of two matrices**
- **Multiplication of two matrices**
- **Exit from the menu**

Ans:-

<u>Code:-</u>

```
import java.util.*;
class Q10_MatrixOperations {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // Predefined 2x2 matrices
        int[][] A = new int[2][2];
        int[][] B = new int[2][2];
        int[][] result = new int[2][2];
        System.out.println("Enter elements of Matrix A (2x2):");
        for (int i = 0; i < 2; i++)
            for (int j = 0; j < 2; j++)
            A[i][j] = sc.nextInt();
        System.out.println("Enter elements of Matrix B (2x2):");
        for (int i = 0; i < 2; i++)
            for (int j = 0; j < 2; j++)
            B[i][j] = sc.nextInt();
        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Addition\n2. Subtraction\n3. Multiplication\n4. Exit");
            System.out.print("Choose an option: ");
            int choice = sc.nextInt();
            switch (choice) {
                case 1:
                    // Matrix addition
                    for (int i = 0; i < 2; i++)
                        for (int j = 0; j < 2; j++)
                            result[i][j] = A[i][j] + B[i][j];
                    break;
                case 2:
                    // Matrix subtraction
                    for (int i = 0; i < 2; i++)
                        for (int j = 0; j < 2; j++)
                            result[i][j] = A[i][j] - B[i][j];
                    break;
                case 3:
                    // Matrix multiplication
                    for (int i = 0; i < 2; i++) {
                        for (int j = 0; j < 2; j++) {
                            result[i][j] = 0;
                            for (int k = 0; k < 2; k++)
                                result[i][j] += A[i][k] * B[k][j];
                        }
                    }
                    break;
                case 4:
                    sc.close();
                    System.exit(0);
            }
            // Display result matrix
            System.out.println("Result:");
            for (int[] row : result) {
                for (int val : row) System.out.print(val + " ");
                System.out.println();
            }
        }
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Q10_MatrixOperations
Enter elements of Matrix A (2x2):
2 4
3 5
Enter elements of Matrix B (2x2):
7 4
8 9

Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Exit
Choose an option: 1
Result:
9 8
11 14

Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Exit
Choose an option: 2
Result:
-5 0
-5 -4

Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Exit
Choose an option: 3
Result:
46 44
61 57

Menu:
1. Addition
2. Subtraction
3. Multiplication
4. Exit
Choose an option: 
```

Assignment_03

Q1. Encapsulation with Bank Account -

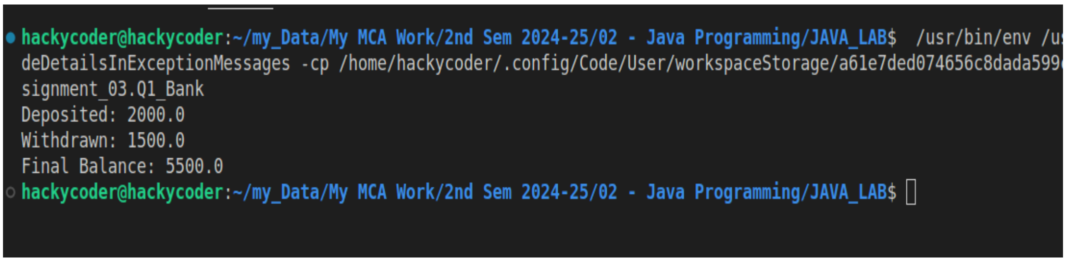
Design a `BankAccount` class to demonstrate encapsulation. The class should store account details (`account number`, `account holder`, `balance`) as **private** data members and provide **getter** and **setter** methods to update and display details. Write a program to **deposit** and **withdraw** money securely.

Ans:-

Code:-

```
package Assignment_03;
class BankAccount {
    private String accountHolder;
    private int accountNumber;
    private double balance;
    public BankAccount(String accountHolder, int accountNumber, double balance) {
        this.accountHolder = accountHolder;
        this.accountNumber = accountNumber;
        this.balance = balance;
    }
    // Getters and Setters
    public String getAccountHolder() {
        return accountHolder;
    }
    public void setAccountHolder(String accountHolder) {
        this.accountHolder = accountHolder;
    }
    public int getAccountNumber() {
        return accountNumber;
    }
    public double getBalance() {
        return balance;
    }
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    }
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance!");
        }
    }
}
public class Q1_Bank {
    public static void main(String[] args) {
        BankAccount acc = new BankAccount("Mohit", 101, 5000);
        acc.deposit(2000);
        acc.withdraw(1500);
        System.out.println("Final Balance: " + acc.getBalance());
    }
}
```

Output:-



```
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /u
deDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599
signment_03.Q1_Bank
Deposited: 2000.0
Withdrawn: 1500.0
Final Balance: 5500.0
○ hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ ▯
```

Q.2 Inheritance & Polymorphism with Shapes

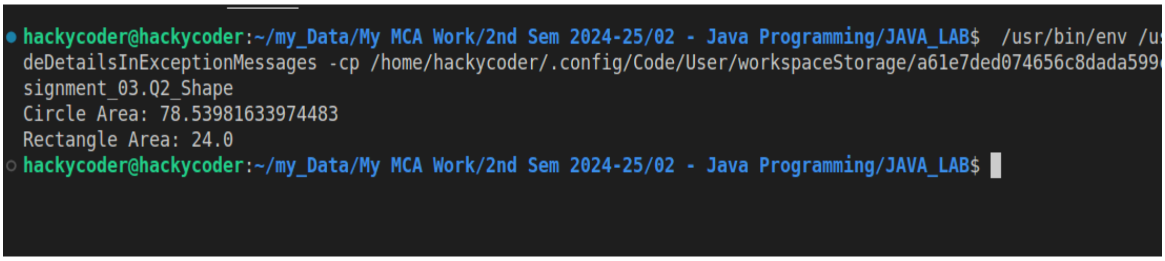
Create a base class `Shape` with a method `area()`. Derive `Circle` and `Rectangle` classes from it, each overriding the `area()` method. Demonstrate **runtime polymorphism by calling `area()` using a `Shape` reference for different objects.**

Ans:-

Code:-

```
package Assignment_03;
abstract class Shape {
    abstract double area();
}
class Circle extends Shape {
    double radius;
    Circle(double r) {
        radius = r;
    }
    double area() {
        return Math.PI * radius * radius;
    }
}
class Rectangle extends Shape {
    double length, breadth;
    Rectangle(double l, double b) {
        length = l;
        breadth = b;
    }
    double area() {
        return length * breadth;
    }
}
public class Q2_Shape {
    public static void main(String[] args) {
        Shape s1 = new Circle(5);
        Shape s2 = new Rectangle(4, 6);
        System.out.println("Circle Area: " + s1.area());
        System.out.println("Rectangle Area: " + s2.area());
    }
}
```

Output:-



```
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/env /u
  deDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8dada599
  signment_03.Q2_Shape
  Circle Area: 78.53981633974483
  Rectangle Area: 24.0
○ hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ ▌
```

Q.3 Constructors and Data Abstraction with Student Details

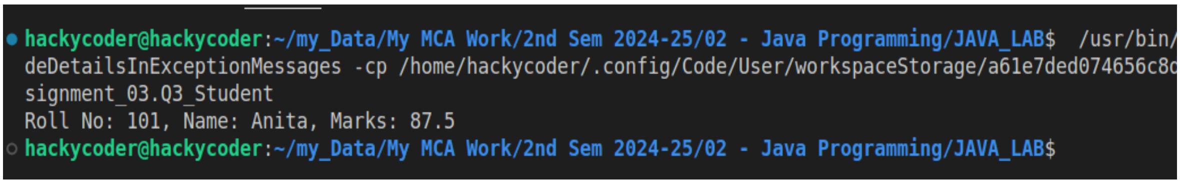
Implement a `Student` class using **data abstraction** that hides details like `roll number`, `name`, and `marks`. Use **parameterized constructors** to initialize values and a **destructor** (`finalize` method) to print a message when the object is destroyed. Demonstrate object creation and destruction in the `main` method.

Ans:-

Code:-

```
package Assignment_03;
class Student {
    private int rollNo;
    private String name;
    private double marks;
    // Constructor
    Student(int r, String n, double m) {
        rollNo = r;
        name = n;
        marks = m;
    }
    void display() {
        System.out.println("Roll No: " + rollNo + ", Name: " + name + ", Marks: " + marks);
    }
    // Destructor simulation
    protected void finalize() {
        System.out.println("Object destroyed for student: " + name);
    }
}
public class Q3_Student {
    public static void main(String[] args) {
        Student s1 = new Student(101, "Anita", 87.5);
        s1.display();
        s1 = null; // Eligible for GC
        System.gc(); // Request GC
    }
}
```

Output:-



```
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/
  deDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded074656c8c
  signment_03.Q3_Student
  Roll No: 101, Name: Anita, Marks: 87.5
○ hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

Q.4 Employee Payroll using Inheritance

Create a base class `Employee` with attributes like `name` and `basic salary`. Derive classes `Manager` and `Programmer` that calculate salary differently using **allowances/deductions. Demonstrate **inheritance** and **polymorphism** by displaying salaries of different employees.**

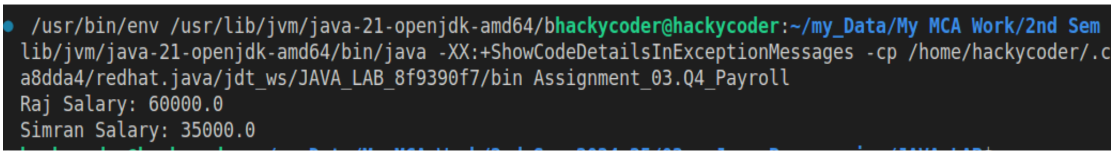
Ans:-

Code:-

```
package Assignment_03;
class Employee {
    String name;
    double basicSalary;

    Employee(String n, double b) {
        name = n;
        basicSalary = b;
    }
    double calculateSalary() {
        return basicSalary;
    }
}
class Manager extends Employee {
    double allowance;
    Manager(String n, double b, double a) {
        super(n, b);
        allowance = a;
    }
    double calculateSalary() {
        return basicSalary + allowance;
    }
}
class Programmer extends Employee {
    double deduction;
    Programmer(String n, double b, double d) {
        super(n, b);
        deduction = d;
    }
    double calculateSalary() {
        return basicSalary - deduction;
    }
}
public class Q4_Payroll {
    public static void main(String[] args) {
        Employee e1 = new Manager("Raj", 50000, 10000);
        Employee e2 = new Programmer("Simran", 40000, 5000);
        System.out.println(e1.name + " Salary: " + e1.calculateSalary());
        System.out.println(e2.name + " Salary: " + e2.calculateSalary());
    }
}
```

Output:-



```
● /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bhackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem
 lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.c
 a8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_03.Q4_Payroll
 Raj Salary: 60000.0
 Simran Salary: 35000.0
```

Q.5 Library Management with Encapsulation

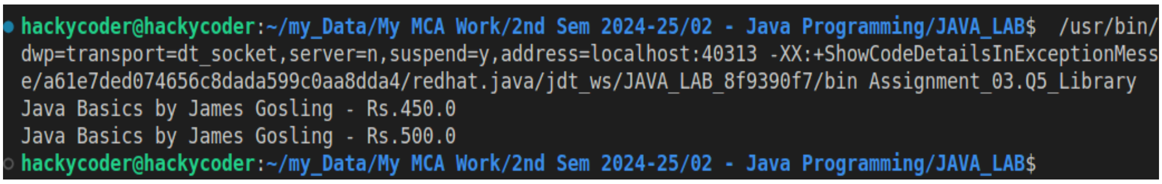
Design a `Book` class that encapsulates book details (`title`, `author`, `price`). Implement **getters and **setters** for updating book information. Create multiple book objects and display their details using a method.**

Ans:-

Code:-

```
package Assignment_03;
class Book {
    private String title;
    private String author;
    private double price;
    public Book(String t, String a, double p) {
        title = t;
        author = a;
        price = p;
    }
    // Getter & Setter
    public String getTitle() {
        return title;
    }
    public void setTitle(String t) {
        title = t;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String a) {
        author = a;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double p) {
        price = p;
    }
    void display() {
        System.out.println(title + " by " + author + " - Rs." + price);
    }
}
public class Q5_Library {
    public static void main(String[] args) {
        Book b1 = new Book("Java Basics", "James Gosling", 450);
        b1.display();
        b1.setPrice(500);
        b1.display();
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /usr/bin/
dwp=transport=dt_socket,server=n,suspend=y,address=localhost:40313 -XX:+ShowCodeDetailsInExceptionMess
e/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_03.Q5_Library
Java Basics by James Gosling - Rs.450.0
Java Basics by James Gosling - Rs.500.0
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

Q.6 Vehicle Hierarchy with Method Overriding

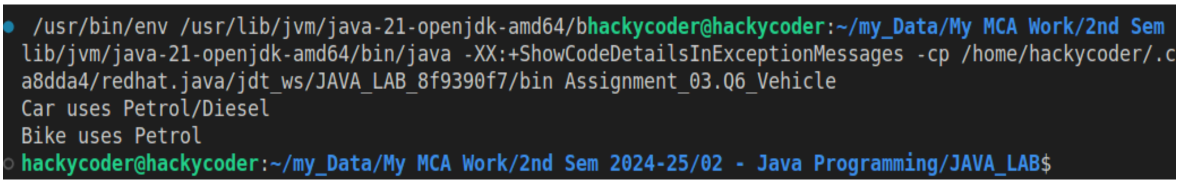
Build a class hierarchy where `Vehicle` is the base class, and `Car` and `Bike` are derived classes. Override a method `fuelType()` in each subclass to demonstrate **runtime polymorphism.**

Ans:-

Code:-

```
package Assignment_03;
class Vehicle {
   void fuelType() {
      System.out.println("Generic fuel type");
   }
}
class Car extends Vehicle {
   void fuelType() {
      System.out.println("Car uses Petrol/Diesel");
   }
}
class Bike extends Vehicle {
   void fuelType() {
      System.out.println("Bike uses Petrol");
   }
}
public class Q6_Vehicle {
   public static void main(String[] args) {
      Vehicle v1 = new Car();
      Vehicle v2 = new Bike();
      v1.fuelType();
      v2.fuelType();
   }
}
```

Output:-



```
 /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bhackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem
lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hackycoder/.c
a8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_03.Q6_Vehicle
Car uses Petrol/Diesel
Bike uses Petrol
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

Q.7 Constructor Overloading for Complex Numbers

Write a class `Complex` that represents complex numbers. Implement **constructor overloading: one **default constructor**, one **parameterized constructor**, and one **copy constructor**. Create objects using all three constructors and display their values.**

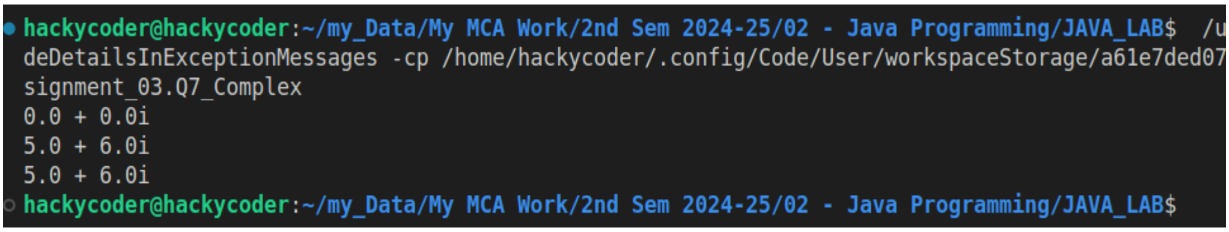
Ans:-

Code:-

```
package Assignment_03;
class Complex {
   double real, imag;
   Complex() { // Default constructor
      real = 0;
      imag = 0;
   }
```

```
    Complex(double r, double i) { // Parameterized constructor
        real = r;
        imag = i;
    }
    Complex(Complex c) { // Copy constructor
        real = c.real;
        imag = c.imag;
    }
    void display() {
        System.out.println(real + " + " + imag + "i");
    }
}
public class Q7_Complex {
    public static void main(String[] args) {
        Complex c1 = new Complex();
        Complex c2 = new Complex(5, 6);
        Complex c3 = new Complex(c2);
        c1.display();
        c2.display();
        c3.display();
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /u
deDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded07
signment_03.Q7_Complex
0.0 + 0.0i
5.0 + 6.0i
5.0 + 6.0i
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

Q.8 Abstract Class for Shape Perimeter

Define an **abstract class `Polygon` with an abstract method `perimeter()`. Implement subclasses `Triangle` and `Square` that provide specific implementations. Demonstrate **data abstraction** and **polymorphism** by calling `perimeter()` through a base class reference.**

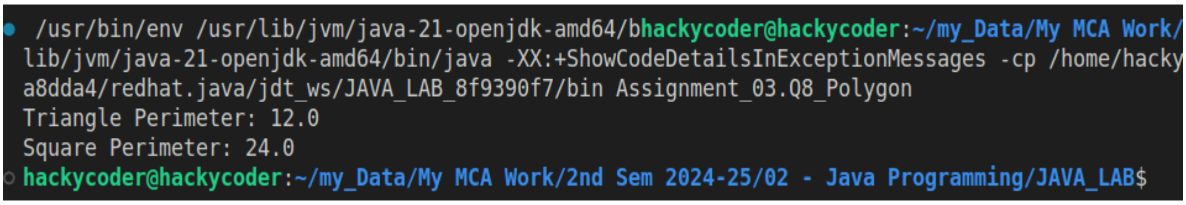
Ans:-

Code:-

```
package Assignment_03;
abstract class Polygon {
    abstract double perimeter();
}
class Triangle extends Polygon {
    double a, b, c;
    Triangle(double x, double y, double z) {
        a = x;
        b = y;
        c = z;
    }
    double perimeter() {
        return a + b + c;
    }
}
```

```
class Square extends Polygon {
    double side;
    Square(double s) {
        side = s;
    }
    double perimeter() {
        return 4 * side;
    }
}
public class Q8_Polygon {
    public static void main(String[] args) {
        Polygon p1 = new Triangle(3, 4, 5);
        Polygon p2 = new Square(6);
        System.out.println("Triangle Perimeter: " + p1.perimeter());
        System.out.println("Square Perimeter: " + p2.perimeter());
    }
}
```

Output:-



```
● /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bhackycoder@hackycoder:~/my_Data/My MCA Work/
 lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /home/hacky
 a8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9390f7/bin Assignment_03.Q8_Polygon
 Triangle Perimeter: 12.0
 Square Perimeter: 24.0
○ hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

Q.9 Student Result with Encapsulation & Constructors

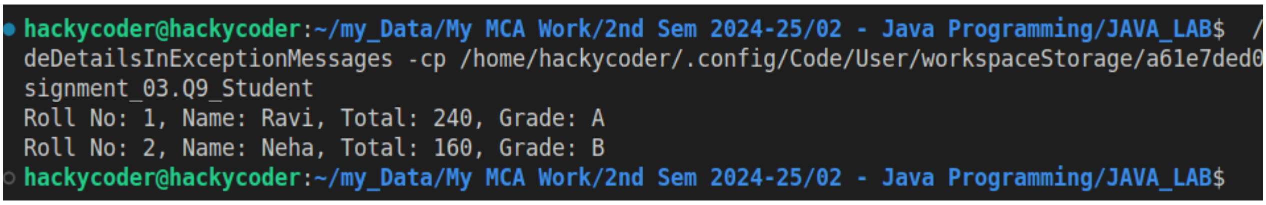
Create a `Student` class with **private attributes** (`roll number`, `name`, `marks in 3 subjects`). Use a **constructor** to initialize the values, and methods to calculate and display the **total** and **grade**. Demonstrate **encapsulation** by providing only controlled access to marks.

Ans:-

Code:-

```
package Assignment_03;
public class Q9_Student {
    private int rollNo;
    private String name;
    private int marks1, marks2, marks3;
    // Constructor
    public Q9_Student(int rollNo, String name, int marks1, int marks2, int marks3) {
        this.rollNo = rollNo;
        this.name = name;
        this.marks1 = marks1;
        this.marks2 = marks2;
        this.marks3 = marks3;
    }
    // Getter methods (controlled access)
    public int getRollNo() {
        return rollNo;
    }
    public String getName() {
        return name;
    }
    // No direct setters for marks, only getters
    public int getMarks1() {
        return marks1;
    }
    public int getMarks2() {
        return marks2;
    }
    public int getMarks3() {
        return marks3;
    }
    // Calculate total marks
    public int getTotal() {
        return marks1 + marks2 + marks3;
    }
    // Calculate grade
    public String getGrade() {
        double avg = getTotal() / 3.0;
        if (avg >= 75)
            return "A";
        else if (avg >= 50)
            return "B";
        else
            return "C";
    }
    // Display student details
    public void display() {
        System.out.println("Roll No: " + rollNo + ", Name: " + name +
                ", Total: " + getTotal() + ", Grade: " + getGrade());
    }
    // Main method to test
    public static void main(String[] args) {
        Q9_Student s1 = new Q9_Student(1, "Ravi", 80, 70, 90);
        Q9_Student s2 = new Q9_Student(2, "Neha", 45, 55, 60);
        s1.display();
        s2.display();
    }
}
```

Output:-



```
● hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /
  deDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e7ded0
  signment_03.Q9_Student
  Roll No: 1, Name: Ravi, Total: 240, Grade: A
  Roll No: 2, Name: Neha, Total: 160, Grade: B
○ hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

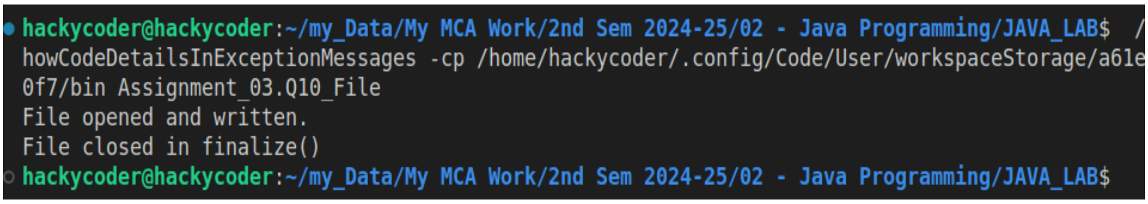
Q.10 Destructor Simulation with File Handling

Ans:-

Code:-

```
package Assignment_03;
import java.io.*;
class FileHandler {
    private BufferedWriter writer;
    FileHandler(String filename) {
        try {
            writer = new BufferedWriter(new FileWriter(filename));
            writer.write("Hello, file handling in Java!");
            System.out.println("File opened and written.");
        } catch (IOException e) {
            System.out.println("Error opening file.");
        }
    }
    protected void finalize() {
        try {
            if (writer != null) {
                writer.close();
                System.out.println("File closed in finalize()");
            }
        } catch (IOException e) {
            System.out.println("Error closing file.");
        }
    }
}
public class Q10_File {
    public static void main(String[] args) {
        new FileHandler("test.txt");
        System.gc();
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /
howCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e
0f7/bin Assignment_03.Q10_File
File opened and written.
File closed in finalize()
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

Q.11 Binary to Decimal Conversion

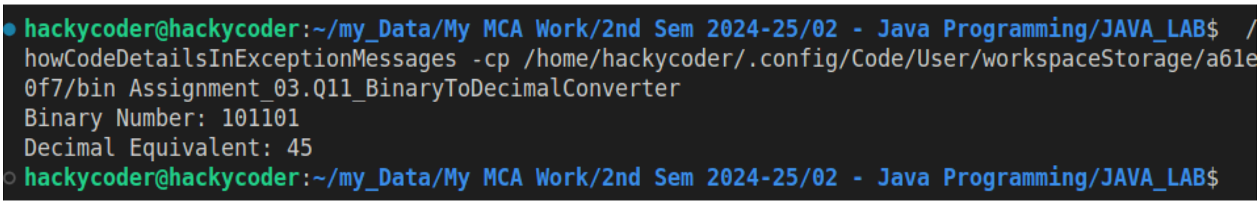
Design a Java program using **Object-Oriented Programming principles to convert a given binary number into its decimal equivalent. The program should demonstrate **encapsulation** by storing the binary number as a private data member, provide **constructors** to initialize the value, and use appropriate methods to perform the conversion. The solution must ensure **data abstraction** by exposing only necessary operations to the user, and should **validate the input** to confirm that only binary digits (0 and 1) are accepted.**

Ans:-

Code:-

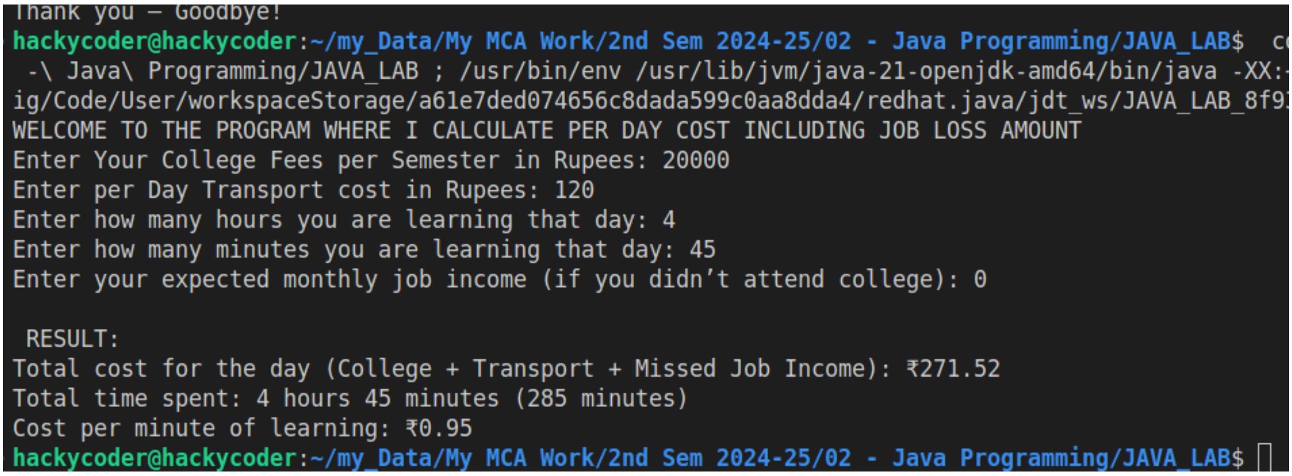
```
package Assignment_03;
// Class to represent a Binary Number
class BinaryNumber {
    // Encapsulation: keeping the data private
    private String binary;
    // Constructor to initialize the binary number
    public BinaryNumber(String binary) {
        if (isValidBinary(binary)) {
            this.binary = binary;
        } else {
            throw new IllegalArgumentException("Invalid binary number! Only 0 and 1 are allowed.");
        }
    }
    // Method to validate if the string is a binary number
    private boolean isValidBinary(String str) {
        return str.matches("[01]+"); // regex ensures only 0s and 1s
    }
    // Method to convert binary to decimal
    public int toDecimal() {
        int decimal = 0;
        int power = 0;
        // Iterate from right to left
        for (int i = binary.length() - 1; i >= 0; i--) {
            char bit = binary.charAt(i);
            if (bit == '1') {
                decimal += Math.pow(2, power);
            }
            power++;
        }
        return decimal;
    }
    // Getter for the binary number (read-only)
    public String getBinary() {
        return binary;
    }
}
// Main class
public class Q11_BinaryToDecimalConverter {
    public static void main(String[] args) {
        try {
            // Creating object using constructor
            BinaryNumber bn = new BinaryNumber("101101");
            // Printing the result
            System.out.println("Binary Number: " + bn.getBinary());
            System.out.println("Decimal Equivalent: " + bn.toDecimal());
        } catch (IllegalArgumentException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Output:-



```
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  /
howCodeDetailsInExceptionMessages -cp /home/hackycoder/.config/Code/User/workspaceStorage/a61e
0f7/bin Assignment_03.Q11_BinaryToDecimalConverter
Binary Number: 101101
Decimal Equivalent: 45
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$
```

Assignment_04



```
Thank you — Goodbye!
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$  c
 -\ Java\ Programming/JAVA_LAB ; /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:
ig/Code/User/workspaceStorage/a61e7ded074656c8dada599c0aa8dda4/redhat.java/jdt_ws/JAVA_LAB_8f9
WELCOME TO THE PROGRAM WHERE I CALCULATE PER DAY COST INCLUDING JOB LOSS AMOUNT
Enter Your College Fees per Semester in Rupees: 20000
Enter per Day Transport cost in Rupees: 120
Enter how many hours you are learning that day: 4
Enter how many minutes you are learning that day: 45
Enter your expected monthly job income (if you didn’t attend college): 0

 RESULT:
Total cost for the day (College + Transport + Missed Job Income): ₹271.52
Total time spent: 4 hours 45 minutes (285 minutes)
Cost per minute of learning: ₹0.95
hackycoder@hackycoder:~/my_Data/My MCA Work/2nd Sem 2024-25/02 - Java Programming/JAVA_LAB$ 
```