

# **Object Oriented Programming**

**(3ITRC2)**

**IT III Semester**

*Submitted by*

**Himanshu Priyadarshi Ahirwar**

**23I3026**

**IT-A**

*Submitted to*

**Aditya Makwe**

Assistant Professor, Department of Information Technology

Institute of Engineering and Technology

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

([www.iet.dauniv.ac.in](http://www.iet.dauniv.ac.in))

**Session July-Dec, 2024**



# ***ASSIGNMENT - 1***

---



**Q1. Write a Java program to change temperature from Celsius to Fahrenheit and vice versa.**

```
package assignment1;
import java.util.Scanner;

public class Question_1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Choose the conversion type:");
        System.out.println("1. Celsius to Fahrenheit");
        System.out.println("2. Fahrenheit to Celsius");

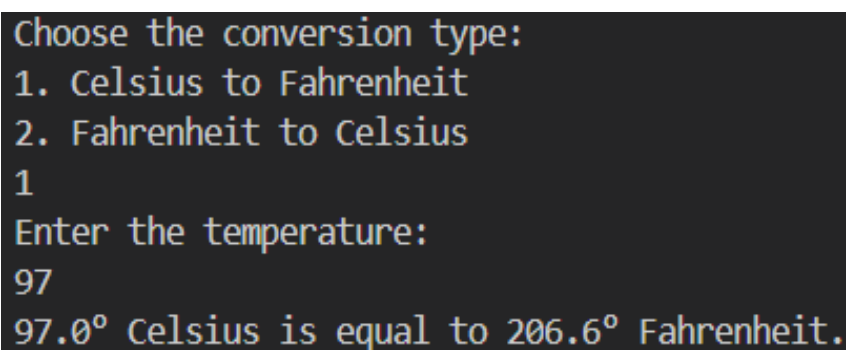
        int choice = scanner.nextInt();

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

        switch (choice) {
            case 1:
                double fahrenheit = (temp * 9 / 5) + 32;
                System.out.println(temp + "° Celsius is equal to " + fahrenheit + "° Fahrenheit.");
                break;
            case 2:
                double celsius = (temp - 32) * 5 / 9;
                System.out.println(temp + "° Fahrenheit is equal to " + celsius + "° Celsius.");
                break;
            default:
                System.out.println("Invalid choice! Please enter 1 or 2.");
                break;
        }

        scanner.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window showing the output of the Java program. The text is as follows:

```
Choose the conversion type:
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
1
Enter the temperature:
97
97.0° Celsius is equal to 206.6° Fahrenheit.
```

## Q2. Write a Java Program to check if a number is Positive or Negative.

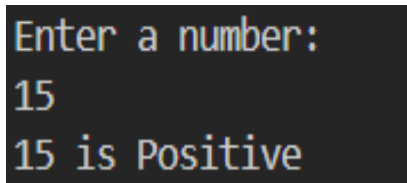
```
package assignment1;
import java.util.Scanner;

public class Question_2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a number:");
        int n = sc.nextInt();

        if (n > 0) {
            System.out.println(n + " is Positive");
        }
        else if (n < 0) {
            System.out.println(n + " is Negative");
        }
        else {
            System.out.println(n + " is Zero");
        }
        sc.close();
    }
}
```

### OUTPUT:-

A screenshot of a terminal window showing the output of the Java program. The text is as follows:

```
Enter a number:
15
15 is Positive
```

**Q3. Write a Java program to find maximum of three numbers.**

```
package assignment1;

import java.util.Scanner;

public class Question_3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

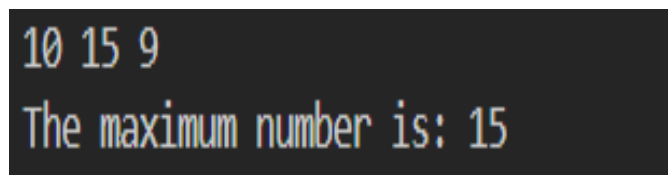
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();

        int max = a;
        if (b > max) {
            max = b;
        }
        if (c > max) {
            max = c;
        }

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

        sc.close();
    }
}
```

**OUTPUT:-**



```
10 15 9
The maximum number is: 15
```

**Q4. Write a Java program to swap two numbers.**

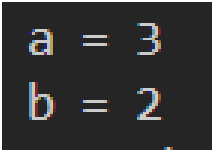
```
package assignment1;

public class Question_4 {
    public static void main(String[] args) {
        int a = 2;
        int b = 3;

        a = a + b;
        b = a - b;
        a = a - b;

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

**OUTPUT:-**



```
a = 3
b = 2
```



**Q5. Write a Java program to convert miles to kilometers.**

```
package assignment1;

import java.util.Scanner;

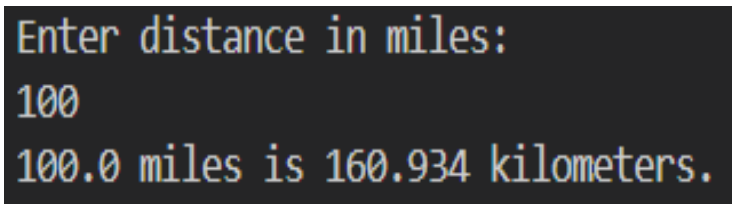
public class Question_5 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

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

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window showing the output of the Java program. The text is displayed in a monospaced font on a dark background. The output consists of three lines: a prompt to enter distance in miles, the user input of 100, and the resulting conversion to 160.934 kilometers.

```
Enter distance in miles:
100
100.0 miles is 160.934 kilometers.
```

**Q6. Write a Java program to check whether a year is leap year or not.**

```
package assignment1;

import java.util.Scanner;

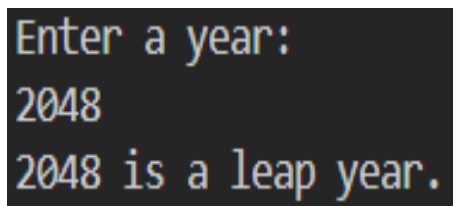
public class Question_6 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

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

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a dark background. It shows the output of the Java program: the prompt "Enter a year:" is followed by the user input "2048", and the final output line is "2048 is a leap year.".

```
Enter a year:
2048
2048 is a leap year.
```

**Q7. Write a Java program for following grading system.Note:**

**Percentage>=90% : Grade A Percentage>=80% : Grade B Percentage>=70% :  
Grade C Percentage>=60% : Grade D Percentage>=40% : Grade E Percentage.**

```
package assignment1;

import java.util.Scanner;

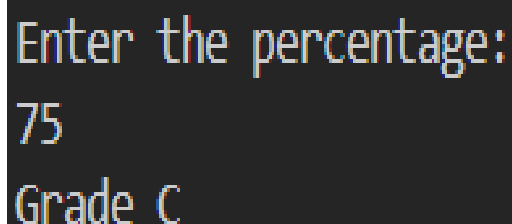
public class Question_7 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

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

        sc.close();
    }
}
```

**OUTPUT:-**



```
Enter the percentage:
75
Grade C
```

**Q8. Write a Java program to check whether a number is divisible by a number given by user.**

```
package assignment1;

import java.util.Scanner;

public class Question_8 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

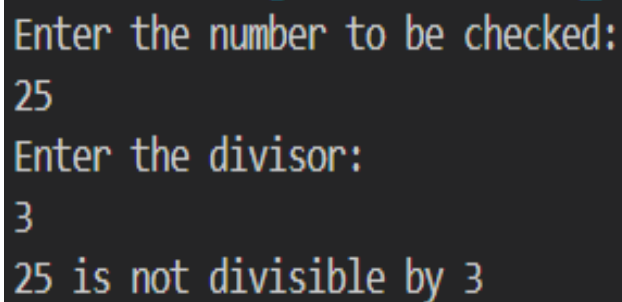
        System.out.println("Enter the number to be checked:");
        int number = sc.nextInt();

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

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

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window showing the output of the Java program. The text is as follows:  
Enter the number to be checked:  
25  
Enter the divisor:  
3  
25 is not divisible by 3  
The text is displayed in a monospaced font on a dark background.

```
Enter the number to be checked:
25
Enter the divisor:
3
25 is not divisible by 3
```

**Q9. Write a Java program to calculate factorial of 12.**

```
package assignment1;

import java.util.Scanner;

public class Question_9 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int number = 12;
        int factorial = 1;

        for (int i = 2; i <= number; i++) {
            factorial *= i;
        }

        System.out.println("Factorial of 12 is: "+factorial);

        sc.close();
    }
}
```

**OUTPUT:-**



```
Factorial of 12 is: 479001600
```

**Q10. Write a Java program for Fibonacci series.**

```
package assignment1;

import java.util.Scanner;

public class Question_10 {
    public static int fibonacci(int n) {
        if (n == 0 || n == 1) {
            return n;
        }
        return fibonacci(n - 1) + fibonacci(n - 2);
    }

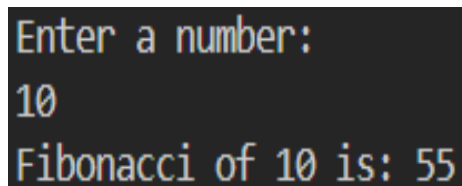
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a number:");
        int n = sc.nextInt();

        System.out.println("Fibonacci of " + n + " is: " + fibonacci(n));

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window showing the output of the Java program. The text is displayed on a dark background with a light-colored monospaced font. The output consists of three lines: 'Enter a number:', '10', and 'Fibonacci of 10 is: 55'.

```
Enter a number:
10
Fibonacci of 10 is: 55
```

**Q11. Write a Java program to reverse a number.**

```
package assignment1;

import java.util.Scanner;

public class Question_11 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

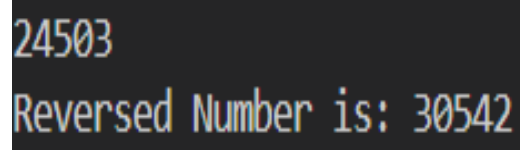
        int n = sc.nextInt();
        int rev = 0;

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

        System.out.println("Reversed Number is: "+rev);

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window showing the output of the Java program. The first line shows the input number '24503' in a light blue font. The second line shows the output 'Reversed Number is: 30542' in a light blue font. The background is dark grey.

```
24503
Reversed Number is: 30542
```

**Q12. Admission to a professional course is subject to the following conditions:**

**(a) marks in Mathematics  $\geq 60$**

**(b) marks in Physics  $\geq 50$**

**(c) marks in Chemistry  $\geq 40$**

**(d) Total in all 3 subjects  $\geq 200$**

**(Or) Total in Maths & Physics  $\geq 150$**

**Given the marks in the 3 subjects of n (user input) students, write a program to**

**Process the applications to list the eligible candidates.**

```
package assignment1;
import java.util.Scanner;

public class Question_12 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter marks for Math, Physics, and Chemistry:");
        int math = sc.nextInt();
        int physics = sc.nextInt();
        int chemistry = sc.nextInt();

        int total = math + physics + chemistry;

        if (math >= 60 && physics >= 50 && chemistry >= 40 && total >= 200) {
            System.out.println("Eligible");
        }
        else if (math + physics >= 150) {
            System.out.println("Eligible");
        }
        else {
            System.out.println("Not Eligible");
        }

        sc.close();
    }
}
```

**OUTPUT:-**

```
Enter marks for Math, Physics, and Chemistry:
85 80 69
Eligible
```



**Q13. Write a Java program to calculate the sum of natural numbers up to a certain range.**

```
package assignment1;

import java.util.Scanner;

public class Question_13 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

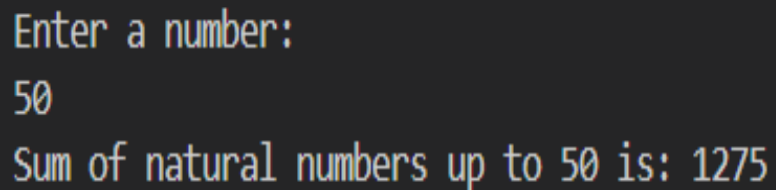
        System.out.println("Enter a number:");
        int n = sc.nextInt();
        int sum = 0;

        for (int i = 1; i <= n; i++) {
            sum += i;
        }

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

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window showing the output of the Java program. The text is displayed in a monospaced font on a dark background. It shows the prompt 'Enter a number:', the user input '50', and the resulting output 'Sum of natural numbers up to 50 is: 1275'.

```
Enter a number:
50
Sum of natural numbers up to 50 is: 1275
```

**Q14. Write a Java program to print all multiple of 10 between a given interval.**

```
package assignment1;

import java.util.Scanner;

public class Question_14 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter start of the interval:");
        int start = sc.nextInt();

        System.out.println("Enter end of the interval:");
        int end = sc.nextInt();

        System.out.println("Multiples of 10 between " + start + " and " + end + ":");
        for (int i = start; i <= end; i++) {
            if (i % 10 == 0) {
                System.out.println(i);
            }
        }

        sc.close();
    }
}
```

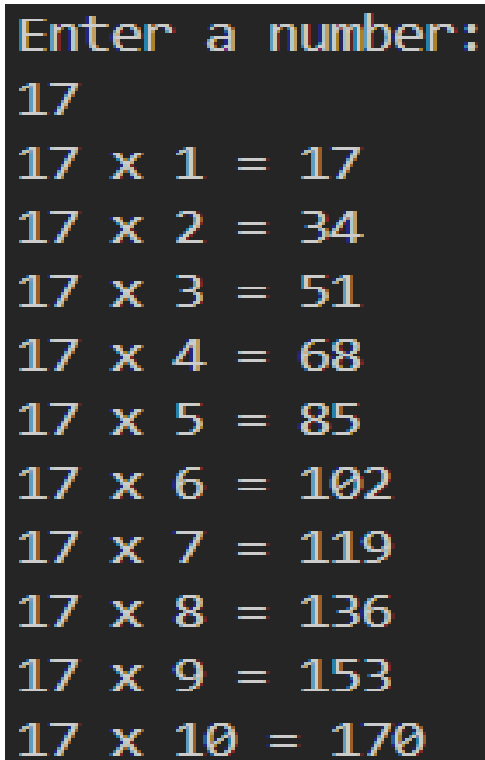
**OUTPUT:-**

```
Enter start of the interval:
10
Enter end of the interval:
100
Multiples of 10 between 10 and 100:
10
20
30
40
50
60
70
80
90
100
```

**Q15. Write a Java program to generate multiplication table.**

```
package assignment1;  
  
import java.util.Scanner;  
  
public class Question_15 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Enter a number:");  
        int n = sc.nextInt();  
  
        for (int i = 1; i <= 10; i++) {  
            System.out.println(n + " x " + i + " = " + (n * i));  
        }  
  
        sc.close();  
    }  
}
```

**OUTPUT:-**



The screenshot shows the output of the Java program. It starts with the prompt "Enter a number:" followed by the user input "17". Below this, the program displays a multiplication table for the number 17, showing the products of 17 with integers from 1 to 10. The output is as follows:

```
Enter a number:  
17  
17 x 1 = 17  
17 x 2 = 34  
17 x 3 = 51  
17 x 4 = 68  
17 x 5 = 85  
17 x 6 = 102  
17 x 7 = 119  
17 x 8 = 136  
17 x 9 = 153  
17 x 10 = 170
```

**Q16. Write a Java program to find HCF of two Numbers.**

```
package assignment1;

import java.util.Scanner;

public class Question_16 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

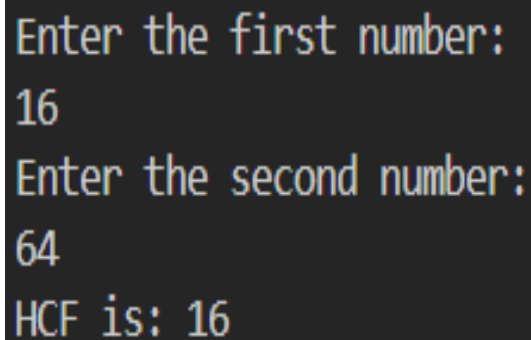
        System.out.println("Enter the second number:");
        int num2 = sc.nextInt();

        while (num1 != num2) {
            if (num1 > num2) {
                num1 = num1 - num2;
            }
            else {
                num2 = num2 - num1;
            }
        }

        System.out.println("HCF is: " + num1);

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a dark background and light-colored text. It shows the execution of the Java program. The first prompt is "Enter the first number:", followed by the input "16". The second prompt is "Enter the second number:", followed by the input "64". The final output line is "HCF is: 16".

```
Enter the first number:
16
Enter the second number:
64
HCF is: 16
```

**Q17. Write a Java program to find LCM of two Numbers.**

```
package assignment1;

import java.util.Scanner;

public class Question_17 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

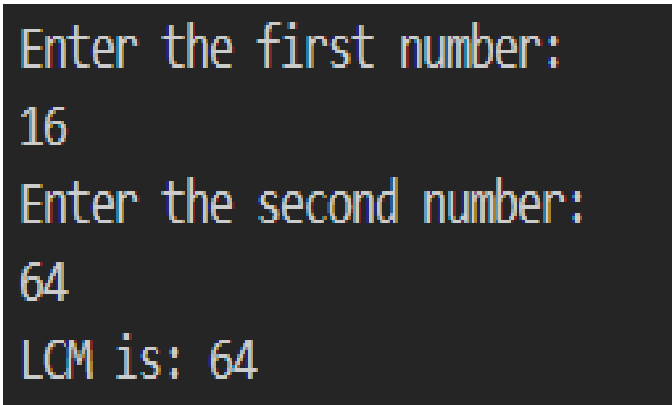
        System.out.println("Enter the second number:");
        int num2 = sc.nextInt();

        int lcm = (num1 > num2) ? num1 : num2;

        while (true) {
            if (lcm % num1 == 0 && lcm % num2 == 0) {
                System.out.println("LCM is: " + lcm);
                break;
            }
            lcm++;
        }

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a dark background. It shows the output of the Java program. The text is as follows:  
Enter the first number:  
16  
Enter the second number:  
64  
LCM is: 64  
The input numbers 16 and 64 are highlighted in yellow in the original image.

```
Enter the first number:
16
Enter the second number:
64
LCM is: 64
```

**Q18. Write a Java program to count the number of digits of an integer.**

```
package assignment1;

import java.util.Scanner;

public class Question_18 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a number:");
        int n = sc.nextInt();

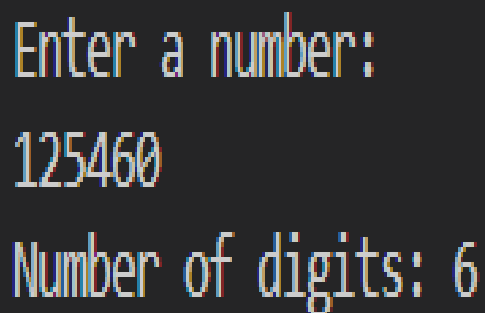
        int count = 0;

        while (n != 0) {
            n /= 10;
            count++;
        }

        System.out.println("Number of digits: " + count);

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a dark background. It shows the output of the Java program. The first line is the prompt "Enter a number:" followed by the user input "125460". The second line is the output "Number of digits: 6".

```
Enter a number:
125460
Number of digits: 6
```

**Q19. Write a Java program to check whether a number is palindrome or not.**

```
package assignment1;

import java.util.Scanner;

public class Question_19 {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Please Enter a number: ");
        int num = sc.nextInt();

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

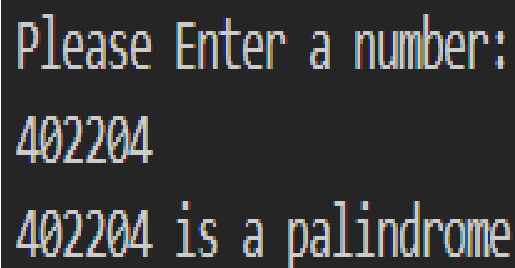
        sc.close();
    }

    public static boolean isPalindrome(int number) {
        int original = number;
        int reversed = 0;

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

        return original == reversed;
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a dark background. It shows the output of the Java program. The first line is the prompt "Please Enter a number:" followed by the user input "402204" on the next line. The third line shows the result "402204 is a palindrome".

```
Please Enter a number:
402204
402204 is a palindrome
```

**Q20. Write a Java program to check whether a number is prime or not.**

```
package assignment1;

import java.util.Scanner;

public class Question_20 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a number:");
        int n = sc.nextInt();

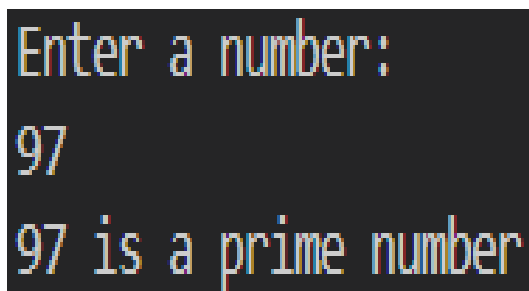
        boolean isPrime = true;

        if (n <= 1) {
            isPrime = false;
        }
        else {
            for (int i = 2; i <= n / 2; i++) {
                if (n % i == 0) {
                    isPrime = false;
                    break;
                }
            }
        }

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

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a dark background. It shows the output of the Java program: the prompt "Enter a number:" is followed by the input "97", and then the output "97 is a prime number".

```
Enter a number:
97
97 is a prime number
```



**Q21. Write a Java program to convert a Binary Number to Decimal and Decimal to Binary.**

```
package assignment1;

import java.util.Scanner;

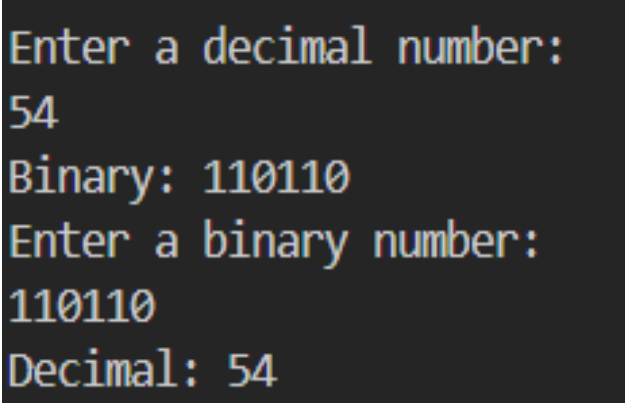
public class Question_21 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a decimal number: ");
        int decimal = sc.nextInt();
        String binary = Integer.toBinaryString(decimal);
        System.out.println("Binary: " + binary);

        System.out.println("Enter a binary number: ");
        String binaryInput = sc.next();
        int decimalEquivalent = Integer.parseInt(binaryInput, 2);
        System.out.println("Decimal: " + decimalEquivalent);

        sc.close();
    }
}
```

**OUTPUT:-**



```
Enter a decimal number:
54
Binary: 110110
Enter a binary number:
110110
Decimal: 54
```

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

```
package assignment1;

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

public class Question_22 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

        int[] arr = new int[n];

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

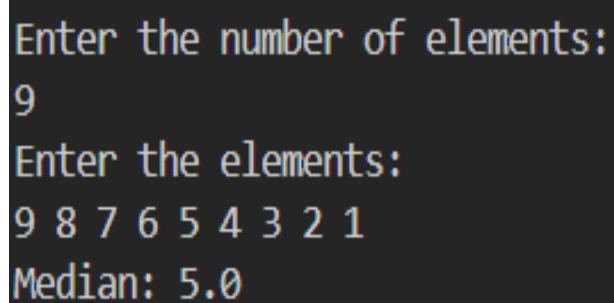
        Arrays.sort(arr);

        double median;

        if (n % 2 == 0) {
            median = (arr[n / 2 - 1] + arr[n / 2]) / 2.0;
        }
        else {
            median = arr[n / 2];
        }

        System.out.println("Median: " + median);
        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window showing the execution of the Java program. The output is as follows:

```
Enter the number of elements:
9
Enter the elements:
9 8 7 6 5 4 3 2 1
Median: 5.0
```

**Q23. Write Java programs for the patterns given bellow:**

**(a)**

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

**(b)**

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

```
package assignment1;
```

```
public class Question_23 {
    public static void main(String[] args) {
        int num = 1;
```

```
        for (int i = 1; i <= 3; i++) {           // Pattern (a)
            for (int j = 1; j <= 2 * i - 1; j++) {
                System.out.print(num++ + " ");
            }
            System.out.println();
        }
```

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

```
        for (int i = 1; i <= 4; i++) {           // Pattern (b)
            for (int j = 1; j <= 4 - i; j++) {
                System.out.print(" ");
            }
            for (int j = i; j >= 1; j--) {
                System.out.print(j + " ");
            }
            for (int j = 2; j <= i; j++) {
                System.out.print(j + " ");
            }
            System.out.println();
        }
    }
}
```

**OUTPUT:-**

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

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

**Q24. Write a Java program to calculate Sum & Average of an integer array.**

```
package assignment1;

import java.util.Scanner;

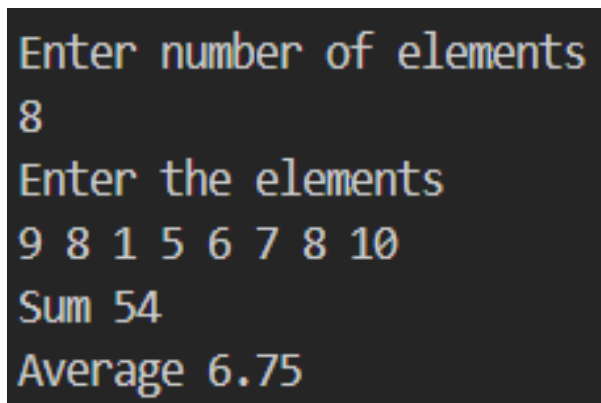
public class Question_24 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of elements ");
        int n = sc.nextInt();
        int[] arr = new int[n];

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

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

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

**OUTPUT:-**

A screenshot of a terminal window showing the output of the Java program. The text is displayed in a monospaced font on a dark background. The output shows the user entering the number of elements as 8, followed by entering the elements 9, 8, 1, 5, 6, 7, 8, and 10. The program then outputs the sum as 54 and the average as 6.75.

```
Enter number of elements
8
Enter the elements
9 8 1 5 6 7 8 10
Sum 54
Average 6.75
```

**Q25. Write a Java program to implement stack using array.**

```
package assignment1;

import java.util.Scanner;

public class Question_25 {
    static int top = -1;
    static int[] stack;

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

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

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

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

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the size of the stack: ");
    int size = sc.nextInt();
    stack = new int[size];

    while (true) {
        System.out.println("Enter 1 to push, 2 to pop, 3 to display, 4 to exit: ");
        int choice = sc.nextInt();

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

## OUTPUT:-

```
Enter the size of the stack:
5
Enter 1 to push, 2 to pop, 3 to display, 4 to exit:
1
Enter element to push:
6
Pushed: 6
Enter 1 to push, 2 to pop, 3 to display, 4 to exit:
1
Enter element to push:
7
Pushed: 7
Enter 1 to push, 2 to pop, 3 to display, 4 to exit:
1
Enter element to push:
8
Pushed: 8
Enter 1 to push, 2 to pop, 3 to display, 4 to exit:
2
Popped: 8
Enter 1 to push, 2 to pop, 3 to display, 4 to exit:
3
Stack elements:
7
6
Enter 1 to push, 2 to pop, 3 to display, 4 to exit:
4
```

**Q26. Write a Java program to implement Queue using array.**

```
package assignment1;

class Queue {
    int[] arr;
    int front;
    int rear;
    int size;
    int capacity;

    Queue(int capacity) {
        this.capacity = capacity;
        arr = new int[capacity];
        front = 0;
        rear = -1;
        size = 0;
    }

    void enqueue(int element) {
        if (size == capacity) {
            System.out.println("Queue is full. Cannot enqueue " + element);
        } else {
            rear = (rear + 1) % capacity;
            arr[rear] = element;
            size++;
            System.out.println("Enqueued: " + element);
        }
    }

    void dequeue() {
        if (size == 0) {
            System.out.println("Queue is empty. Cannot dequeue.");
        } else {
            System.out.println("Dequeued: " + arr[front]);
            front = (front + 1) % capacity;
            size--;
        }
    }

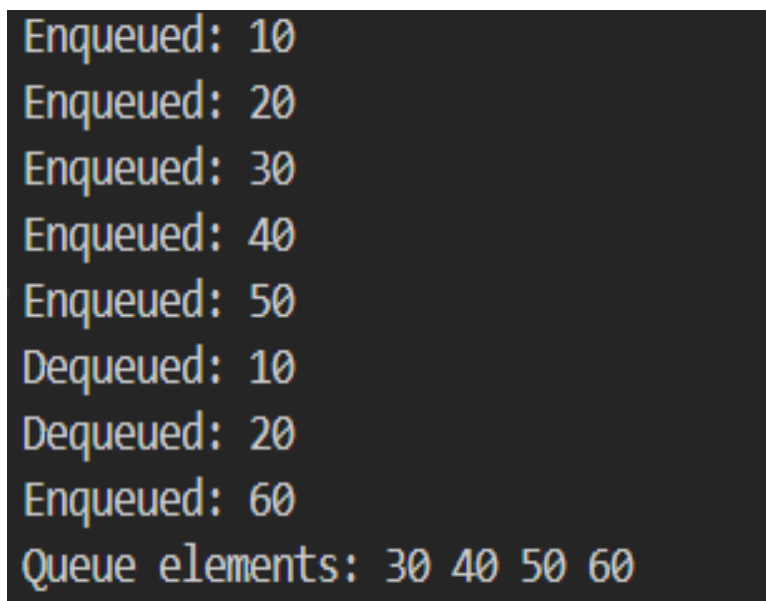
    void display() {
        if (size == 0) {
            System.out.println("Queue is empty");
        } else {
            System.out.print("Queue elements: ");
            int i = front;
```



```
        for (int j = 0; j < size; j++) {  
            System.out.print(arr[i] + " ");  
            i = (i + 1) % capacity;  
        }  
        System.out.println();  
    }  
}
```

```
public class Question_26 {  
    public static void main(String[] args) {  
        Queue queue = new Queue(5);  
  
        queue.enqueue(10);  
        queue.enqueue(20);  
        queue.enqueue(30);  
        queue.enqueue(40);  
        queue.enqueue(50);  
  
        queue.dequeue();  
        queue.dequeue();  
  
        queue.enqueue(60);  
  
        queue.display();  
    }  
}
```

#### OUTPUT:-

A screenshot of a terminal window with a dark background. The output text is displayed in a light blue/cyan monospaced font. The text shows the sequence of operations on a queue: five enqueue operations (10, 20, 30, 40, 50), two dequeue operations (removing 10 and 20), one enqueue operation (60), and a final display operation showing the remaining elements 30, 40, 50, and 60.

```
Enqueued: 10  
Enqueued: 20  
Enqueued: 30  
Enqueued: 40  
Enqueued: 50  
Dequeued: 10  
Dequeued: 20  
Enqueued: 60  
Queue elements: 30 40 50 60
```

**Q27. Write a Java program to enter n elements in an array and find smallest number among them.**

```
package assignment1;

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

public class Question_27 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter number of elements ");
        int n = sc.nextInt();

        int[] arr = new int[n];

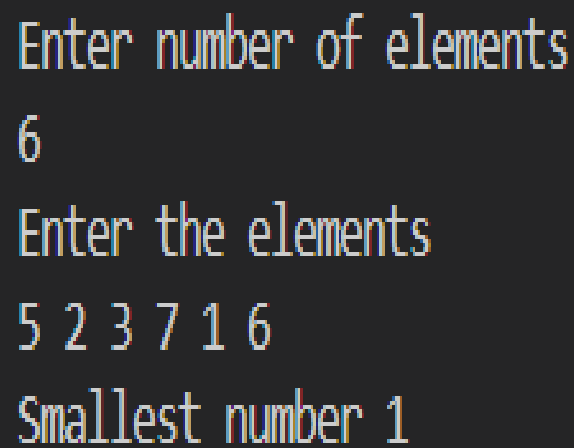
        System.out.println("Enter the elements ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        Arrays.sort(arr);

        System.out.println("Smallest number " + arr[0]);

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a dark background and light-colored text. The output shows the program's execution: it prompts for the number of elements, receives '6', prompts for the elements, receives '5 2 3 7 1 6', and finally outputs 'Smallest number 1'.

```
Enter number of elements
6
Enter the elements
5 2 3 7 1 6
Smallest number 1
```

**Q28. Write Java program to find the sum of all odd numbers in a array.**

```
package assignment1;

import java.util.Scanner;

public class Question_28 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];

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

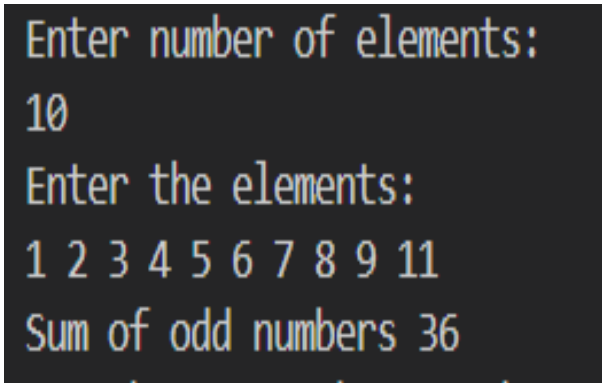
        int sum = 0;
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] % 2 != 0) {

                sum = sum + arr[i];
            }
        }

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

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window showing the execution of the Java program. The text is displayed in a monospaced font on a dark background. The output shows the program prompting for the number of elements, receiving 10, then prompting for the elements, receiving the sequence 1 2 3 4 5 6 7 8 9 11, and finally displaying the sum of odd numbers as 36.

```
Enter number of elements:
10
Enter the elements:
1 2 3 4 5 6 7 8 9 11
Sum of odd numbers 36
```

**Q29. Write a Java program to find duplicate elements in a 1D array and find their frequency of occurrence.**

```
package assignment1;
import java.util.Scanner;

public class Question_29 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];

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

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

        for (int i = 0; i < n; i++) {
            int count = 1;
            if (arr[i] != -1) {

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

**OUTPUT:-**

```
Enter number of elements:
8
Enter the elements:
1 2 6 2 7 2 6 6
Duplicate elements and their frequency:
2 occurs 3 times
6 occurs 3 times
```

**Q30. Write a Java program to print every alternate number of a given array.**

```
package assignment1;
import java.util.Scanner;

public class Question_30 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];

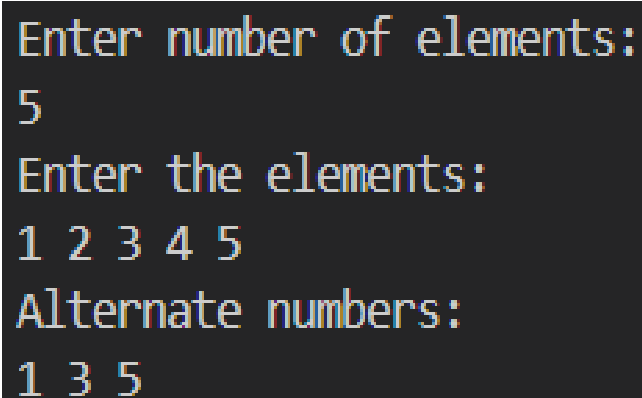
        System.out.println("Enter the elements: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        System.out.println("Alternate numbers:");
        for (int i = 0; i < arr.length; i += 2) {
            System.out.print(arr[i] + " ");
        }

        System.out.println();

        sc.close();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window showing the output of the Java program. The text is as follows:  
Enter number of elements:  
5  
Enter the elements:  
1 2 3 4 5  
Alternate numbers:  
1 3 5  
The text is displayed in a monospaced font with a light blue/cyan color on a black background.

```
Enter number of elements:
5
Enter the elements:
1 2 3 4 5
Alternate numbers:
1 3 5
```

**Q31. Write a Java program to show 0-arguments constructor.**

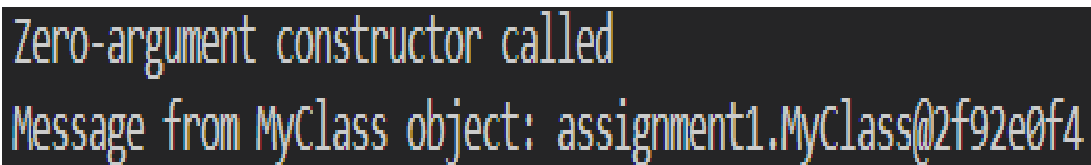
```
package assignment1;

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

    public void displayMessage() {
        System.out.println("Message from MyClass object: " + this);
    }
}

public class Question_31 {
    public static void main(String[] args) {
        MyClass obj = new MyClass();
        obj.displayMessage();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window showing the output of the Java program. The text is displayed in a monospaced font on a dark background. The first line is "Zero-argument constructor called" and the second line is "Message from MyClass object: assignment1.MyClass@2f92e0f4".

```
Zero-argument constructor called
Message from MyClass object: assignment1.MyClass@2f92e0f4
```

**Q32. Write a Java program to show parameterized constructor.**

```
package assignment1;

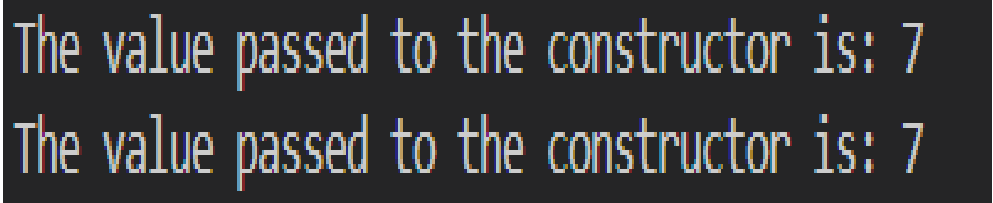
class FirstClass {
    int value;

    public FirstClass(int value) {
        this.value = value;
        display();
    }

    public void display() {
        System.out.println("The value passed to the constructor is: " + value);
    }
}

public class Question_32 {
    public static void main(String[] args) {
        FirstClass obj = new FirstClass(7);
        obj.display();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a black background and light blue text. It displays two identical lines of output: "The value passed to the constructor is: 7".

```
The value passed to the constructor is: 7
The value passed to the constructor is: 7
```

**Q33. Write a class, Commission, which has an instance variable, sales; an appropriate constructor; and a method, commission() that returns the commission. Now write a demo class to test the Commission class by reading a sale from the user, using it to create a Commission object after validating that the value is not negative. Finally, call the commission() method to get and print the commission. If the sales are negative, your demo should print the message "Invalid Input".**

```
package assignment1;
```

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

```
class Commission {
```

```
    private double sales;
```

```
    public Commission(double sales) {
```

```
        if (sales < 0) {
```

```
            System.out.println("Invalid Input");
```

```
            this.sales = 0;
```

```
        }
```

```
        else {
```

```
            this.sales = sales;
```

```
        }
```

```
    }
```

```
    public double calculateCommission() {
```

```
        return sales * 0.1;
```

```
    }
```

```
}
```

```
public class Question_33 {
```

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

```
        Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter sales amount: ");
```

```
        double sales = sc.nextDouble();
```

```
        Commission commissionObj = new Commission(sales);
```

```
        if (sales >= 0) {
```

```
            System.out.println("Commission: " + commissionObj.calculateCommission());
```

```
        }
```

```
        else {
```

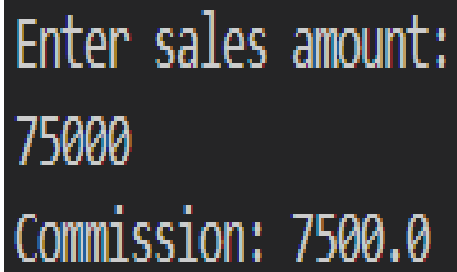
```
            System.out.println("Commission not calculated due to invalid input.");
```

```
        }
```



```
        sc.close();  
    }  
}
```

**OUTPUT:-**

A screenshot of a terminal window with a black background and light blue text. It shows the output of a program where the user enters a sales amount of 75000, and the program calculates a commission of 7500.0.

```
Enter sales amount:  
75000  
Commission: 7500.0
```



# ***ASSIGNMENT - 2***

---



**Q1. Given are two one-dimensional arrays A & B, which are sorted in ascending order. Write a Java program to merge them into single sorted array C that contains every item from arrays A & B, in ascending order.**

```
package assignment2;

public class Question1 {
    public static void main(String[] args) {
        int A[] = { 1, 3, 5, 7, 9 };
        int B[] = { 2, 4, 6, 8 };

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

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

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

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

        for(int x:C){
            System.out.print(x + " ");
        }
    }
}
```

**OUTPUT:-**



**Q2. Write a Java program to show 0-arguments constructor.**

```
package assignment1;

public class Question2 {

    Question2() {
        System.out.println("This is a 0-argument constructor");
    }

    public static void main(String[] args) {
        new Question2();
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a dark background. The text "This is a 0-argument constructor" is displayed in a light blue, monospaced font.

**Q3. Write a Java program to show parameterized constructor.**

```
package assignment1;

public class Question3 {
    int number;

    Question3(int number) {
        this.number = number;
    }

    public static void main(String[] args) {
        Question3 obj = new Question3(10);
        System.out.println("Number = " + obj.number);
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a dark background. The text "Number = 10" is displayed in a light blue, monospaced font. The text is centered horizontally and appears to be the output of a Java program.

#### Q4. Write a Java program to show constructor overloading.

```
package assignment1;

class Example {
    int x;
    int y;

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

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

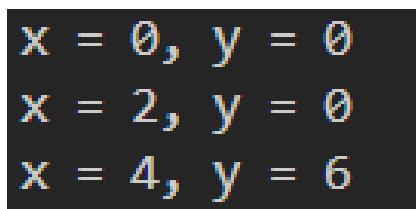
    Example(int a, int b) {
        x = a;
        y = b;
    }

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

public class Question4 {
    public static void main(String[] args) {
        Example obj1 = new Example();
        Example obj2 = new Example(2);
        Example obj3 = new Example(4, 6);

        obj1.display();
        obj2.display();
        obj3.display();
    }
}
```

**OUTPUT:-**



```
x = 0, y = 0
x = 2, y = 0
x = 4, y = 6
```



**Q5. Write a Java program to implement the concept of inheritance.**

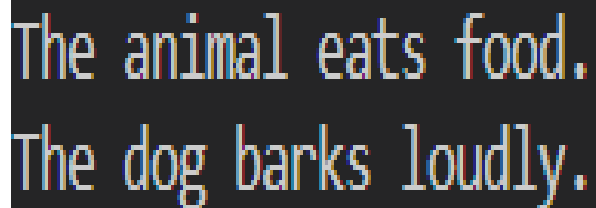
```
package assignment1;

public class Question5 {
    public static void main(String[] args) {
        Dog myDog = new Dog();
        myDog.eatFood();
        myDog.makeSound();
    }
}

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

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

**OUTPUT:-**



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

**Q6. Write a Java program to show method overloading.**

```
package assignment1;

public class Question6 {
    int sum(int num1, int num2) {
        return num1 + num2;
    }

    int sum(int num1, int num2, int num3) {
        return num1 + num2 + num3;
    }

    public static void main(String[] args) {
        Question6 obj = new Question6();
        int result1 = obj.sum(5, 7);
        int result2 = obj.sum(5, 7, 9);

        System.out.println(result1);
        System.out.println(result2);
    }
}
```

**OUTPUT:-**



```
12
21
```

**Q7. Write a Java program to show method overriding.**

```
package assignment1;

public class Question7 {
    public static void main(String[] args) {
        Student student = new Student();
        student.show();
    }
}

class Teacher {
    void show() {
        System.out.println("This is the teacher class.");
    }
}

class Student extends Teacher {
    void show() {
        System.out.println("This is the student class.");
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a black background. The text "This is the student class." is displayed in a yellow, monospaced font. The text is slightly blurred, suggesting it was captured from a video or a fast-moving screen.

**Q8. Write a Java program to show method hiding.**

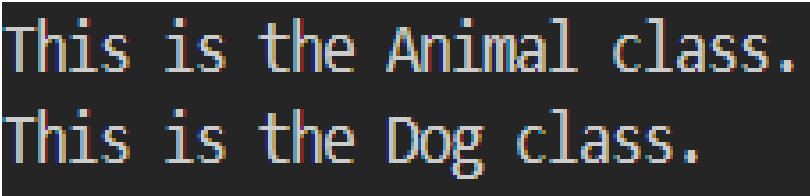
```
package assignment1;

public class Question8 {
    public static void main(String[] args) {
        Animal.show();
        Dog.show();
    }

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

    static class Dog extends Animal {
        static void show() {
            System.out.println("This is the Dog class.");
        }
    }
}
```

**OUTPUT:-**



The screenshot shows the output of the Java program in a monospaced font on a dark background. The output consists of two lines: "This is the Animal class." followed by "This is the Dog class." on the next line. The text is displayed in a light blue/cyan color.

**Q9. Create a general class ThreeDObject and derive the classes Box, Cube, Cylinder and Cone from it. The class ThreeDObject has methods wholeSurfaceArea ( ) and volume ( ). Override these two methods in each of the derived classes to calculate the volume and whole surface area of each type of three-dimensional objects. The dimensions of the objects are to be taken from the users and passed through the respective constructors of each derived class. Write a main method to test these classes.**

```
package assignment1;
```

```
public class Question9 {  
    public static void main(String[] args) {  
        Rectangle box = new Rectangle(3, 4, 5);  
        Square cube = new Square(3);  
        Tube cylinder = new Tube(3, 5);  
        Pyramid cone = new Pyramid(3, 5);  
  
        System.out.println("Rectangle - Volume: " + box.volume() + ", Surface Area: " + box.surfaceArea());  
        System.out.println("Square - Volume: " + cube.volume() + ", Surface Area: " + cube.surfaceArea());  
        System.out.println("Tube - Volume: " + cylinder.volume() + ", Surface Area: " + cylinder.surfaceArea());  
        System.out.println("Pyramid - Volume: " + cone.volume() + ", Surface Area: " + cone.surfaceArea());  
    }  
}
```

```
abstract class Shape3D {  
    abstract double surfaceArea();  
    abstract double volume();  
}
```

```
class Rectangle extends Shape3D {  
    double length, width, height;
```

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

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

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

```
}
```

```
class Square extends Shape3D {
```

```
    double side;
```

```
    Square(double s) {
```

```
        side = s;
```

```
    }
```

```
    double surfaceArea() {
```

```
        return 6 * side * side;
```

```
    }
```

```
    double volume() {
```

```
        return side * side * side;
```

```
    }
```

```
}
```

```
class Tube extends Shape3D {
```

```
    double radius, height;
```

```
    Tube(double r, double h) {
```

```
        radius = r;
```

```
        height = h;
```

```
    }
```

```
    double surfaceArea() {
```

```
        return 2 * Math.PI * radius * (radius + height);
```

```
    }
```

```
    double volume() {
```

```
        return Math.PI * radius * radius * height;
```

```
    }
```

```
}
```

```
class Pyramid extends Shape3D {
```

```
    double radius, height;
```

```
    Pyramid(double r, double h) {
```

```
        radius = r;
```

```
        height = h;
```

```
    }
```

```
    double surfaceArea() {
```

```
        double slantHeight = Math.sqrt(radius * radius + height * height);
```

```
        return Math.PI * radius * (radius + slantHeight);
```

```
    }
```

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

#### OUTPUT:-

```
Rectangle - Volume: 60.0, Surface Area: 94.0  
Square - Volume: 27.0, Surface Area: 54.0  
Tube - Volume: 141.3716694115407, Surface Area: 150.79644737231007  
Pyramid - Volume: 47.1238898038469, Surface Area: 83.22976079115259
```

**Q10. Write a program to create a class named Vehicle having protected instance variables regnNumber, speed, color, ownerName and a method showData ( ) to show “This is a vehicle class”. Inherit the Vehicle class into subclasses named Bus and Car having individual private instance variables routeNumber in Bus and manufacturerName in Car and both of them having showData ( ) method showing all details of Bus and Car respectively with content of the super class’s showData ( ) method.**

```
package assignment1;
```

```
public class Question10 {
    public static void main(String[] args) {
        Bus bus1 = new Bus("B123", 60, "Yellow", "Alice", "Route 5");
        Car car1 = new Car("C456", 150, "Black", "Bob", "Honda");

        bus1.showDetails();
        car1.showDetails();
    }
}

class Vehicle {
    protected String registrationNumber;
    protected int maxSpeed;
    protected String vehicleColor;
    protected String owner;

    Vehicle(String registrationNumber, int maxSpeed, String vehicleColor, String owner) {
        this.registrationNumber = registrationNumber;
        this.maxSpeed = maxSpeed;
        this.vehicleColor = vehicleColor;
        this.owner = owner;
    }

    void showDetails() {
        System.out.println("This is a general vehicle.");
    }
}

class Bus extends Vehicle {
    private String busRoute;

    Bus(String registrationNumber, int maxSpeed, String vehicleColor, String owner, String busRoute) {
        super(registrationNumber, maxSpeed, vehicleColor, owner);
        this.busRoute = busRoute;
    }
}
```



```

void showDetails() {
    super.showDetails(); // Calls the method from the Vehicle class
    System.out.println("Bus Registration: " + registrationNumber + ", Speed: " + maxSpeed +
        " km/h, Color: " + vehicleColor + ", Owner: " + owner + ", Route: " + busRoute);
}
}

class Car extends Vehicle {
    private String carManufacturer;

    Car(String registrationNumber, int maxSpeed, String vehicleColor, String owner, String carManufacturer)
    {
        super(registrationNumber, maxSpeed, vehicleColor, owner);
        this.carManufacturer = carManufacturer;
    }

    void showDetails() {
        super.showDetails(); // Calls the method from the Vehicle class
        System.out.println("Car Registration: " + registrationNumber + ", Speed: " + maxSpeed +
            " km/h, Color: " + vehicleColor + ", Owner: " + owner + ", Manufacturer: " + carManufacturer);
    }
}

```

### OUTPUT:-

```

This is a general vehicle.
Bus Registration: B123, Speed: 60 km/h, Color: Yellow, Owner: Alice, Route: Route 5
This is a general vehicle.
Car Registration: C456, Speed: 150 km/h, Color: Black, Owner: Bob, Manufacturer: Honda

```

**Q11. Write a Java program which creates a base class Num and contains an integer number along with a method shownum() which displays the number. Now create a derived class HexNum which inherits Num and overrides shownum() which displays the hexadecimal value and octal value of the number. Demonstrate the working of the classes.**

```
package assignment1;

public class Question11 {
    public static void main(String[] args) {
        Number num1 = new Number(56);
        AdvancedNumber num2 = new AdvancedNumber(56);

        num1.displayNum();
        num2.displayNum();
    }
}

class Number {
    int value;

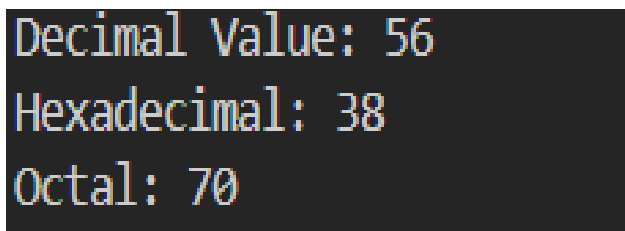
    Number(int value) {
        this.value = value;
    }

    void displayNum() {
        System.out.println("Decimal Value: " + value);
    }
}

class AdvancedNumber extends Number {
    AdvancedNumber(int value) {
        super(value);
    }

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

**OUTPUT:-**



```
Decimal Value: 56
Hexadecimal: 38
Octal: 70
```

**Q12. Create a base class Distance which stores the distance between two locations in miles and a method travelTime(). The method prints the time taken to cover the distance when the speed is 60 miles per hour. Now in a derived class DistanceMKS, override travelTime() so that it prints the time assuming the distance is in kilometers and the speed is 100 km per second. Demonstrate the working of the classes.**

```
package assignment1;

public class Question12 {
    public static void main(String[] args) {
        Journey journey1 = new Journey(120);
        JourneyMetricSystem journey2 = new JourneyMetricSystem(120);

        journey1.calculateTime();
        journey2.calculateTime();
    }
}

class Journey {
    double distanceInMiles;

    Journey(double distanceInMiles) {
        this.distanceInMiles = distanceInMiles;
    }

    void calculateTime() {
        System.out.println("Time to cover the distance (in hours) with speed 60 mph: " +
            (distanceInMiles / 60));
    }
}

class JourneyMetricSystem extends Journey {
    JourneyMetricSystem(double distanceInKilometers) {
        super(distanceInKilometers);
    }

    void calculateTime() {
        System.out.println("Time to cover the distance (in hours) with speed 100 km/s: " +
            + (distanceInMiles / 100));
    }
}
```

**OUTPUT:-**

```
Time to cover the distance (in hours) with speed 60 mph: 2.0
Time to cover the distance (in hours) with speed 100 km/s: 1.2
```

**Q13. Write a Java program to explain “multilevel inheritance.”**

```
package assignment1;

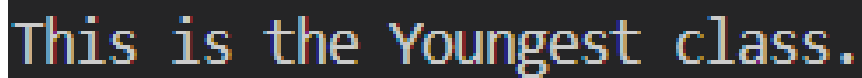
public class Question13 {
    public static void main(String[] args) {
        Youngest childObject = new Youngest();
        childObject.display();
    }
}

class Ancestor {
    void display() {
        System.out.println("This is the Ancestor class.");
    }
}

class ParentGeneration extends Ancestor {
    void display() {
        System.out.println("This is the Parent Generation class.");
    }
}

class Youngest extends ParentGeneration {
    void display() {
        System.out.println("This is the Youngest class.");
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a black background. The text "This is the Youngest class." is displayed in a light blue, monospaced font. The text is centered horizontally and appears to be the output of the Java program shown in the previous block.

**Q14. Write a program to define a class Employee to accept emp\_id, emp\_name, basic\_salary from the user and display the gross\_salary.**

```
package assignment1;
import java.util.Scanner;

public class Question14 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter Employee ID: ");
        int empId = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Enter Employee Name: ");
        String empName = scanner.nextLine();
        System.out.print("Enter Basic Salary: ");
        double basicSalary = scanner.nextDouble();

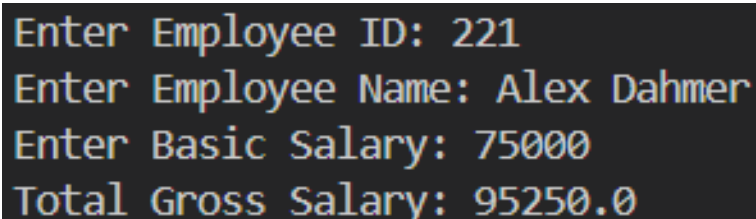
        Worker worker = new Worker(empId, empName, basicSalary);
        worker.showGrossSalary();
        scanner.close();
    }
}

class Worker {
    int id;
    String name;
    double basicSalary;

    Worker(int id, String name, double salary) {
        this.id = id;
        this.name = name;
        this.basicSalary = salary;
    }

    void showGrossSalary() {
        double grossSalary = basicSalary + (basicSalary * 0.15) + (basicSalary * 0.12);
        System.out.println("Total Gross Salary: " + grossSalary);
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window with a dark background. It shows the output of the Java program. The text is as follows:  
Enter Employee ID: 221  
Enter Employee Name: Alex Dahmer  
Enter Basic Salary: 75000  
Total Gross Salary: 95250.0

**Q15. Write a program to demonstrate use of 'this' keyword.**

```
package assignment1;

public class Question15 {
    public static void main(String[] args) {
        NumberDemo demoObject = new NumberDemo(25);
        demoObject.display();
    }
}

class NumberDemo {
    int number;

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

    void display() {
        System.out.println("The number is: " + this.number);
    }
}
```

**OUTPUT:-**

A screenshot of a terminal window showing the output of the Java program. The text "The number is: 25" is displayed in a monospaced font with a light blue/cyan color on a black background. There are small blue and red marks above the text, possibly indicating cursor positions or line numbers.

The number is: 25

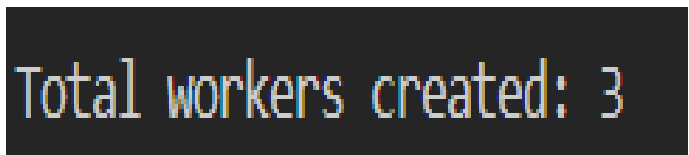
**Q16. Write a program to demonstrate use of 'static' keyword.**

```
package assignment1;
```

```
public class Question16 {  
    public static void main(String[] args) {  
        new Workers();  
        new Workers();  
        new Workers();  
  
        Workers.showTotalWorkers();  
    }  
}
```

```
class Workers {  
    static int totalWorkers = 0;  
  
    Workers() {  
        totalWorkers++;  
    }  
  
    // Static method  
    static void showTotalWorkers() {  
        System.out.println("Total workers created: " + totalWorkers);  
    }  
}
```

**OUTPUT:-**



```
Total workers created: 3
```

**Q17. Write program, which finds the sum of numbers formed by consecutive digits.**

**Input : 2415**

**Output : 24+41+15=80.**

```
package assignment1;
```

```
public class Question17 {  
    public static void main(String[] args) {  
        int number = 2415;  
        String numStr = Integer.toString(number);  
        int total = 0;  
  
        for (int i = 0; i < numStr.length() - 1; i++) {  
            String pair = numStr.substring(i, i + 2);  
            int value = Integer.parseInt(pair);  
            total += value;  
        }  
  
        System.out.println("The sum of consecutive numbers is: " + total);  
    }  
}
```

**OUTPUT:-**



```
The sum of consecutive numbers is: 80
```



**Q18. Create three interfaces, each with two methods. Inherit a new interface from the three, adding a new method. Create a class by implementing the new interface and also inheriting from a concrete class. Now write four methods, each of which takes one of the four interfaces as an argument. In main ( ), create an object of your class and pass it to each of the methods.**

```
package assignment1;
```

```
public class Question18 {  
    public static void main(String[] args) {  
        AllMethods obj = new AllMethods();  
        obj.taskOne();  
        obj.taskTwo();  
        obj.taskThree();  
        obj.finalTask();  
        obj.showConcreteMethod();  
    }  
}
```

```
interface FirstInterface {  
    void taskOne();  
    void taskTwo();  
}
```

```
interface SecondInterface {  
    void taskThree();  
    void taskFour();  
}
```

```
interface ThirdInterface {  
    void taskFive();  
    void taskSix();  
}
```

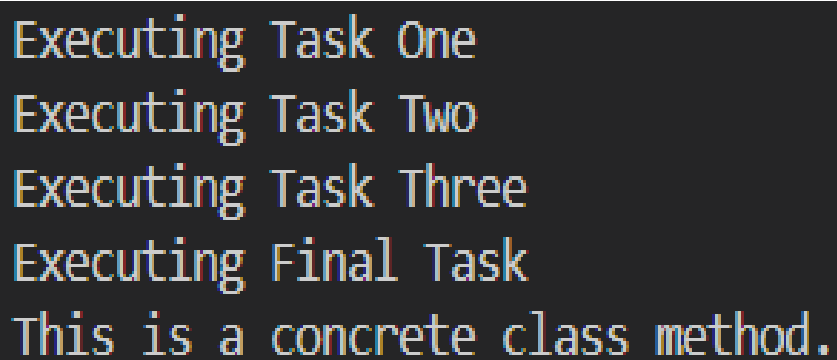
```
interface AllInterfaces extends FirstInterface, SecondInterface, ThirdInterface {  
    void finalTask();  
}
```

```
class BaseClass {  
    void showConcreteMethod() {  
        System.out.println("This is a concrete class method.");  
    }  
}
```

```
class AllMethods extends BaseClass implements AllInterfaces {
```

```
public void taskOne() {  
    System.out.println("Executing Task One");  
}  
  
public void taskTwo() {  
    System.out.println("Executing Task Two");  
}  
  
public void taskThree() {  
    System.out.println("Executing Task Three");  
}  
  
public void taskFour() {  
    System.out.println("Executing Task Four");  
}  
  
public void taskFive() {  
    System.out.println("Executing Task Five");  
}  
  
public void taskSix() {  
    System.out.println("Executing Task Six");  
}  
  
public void finalTask() {  
    System.out.println("Executing Final Task");  
}  
}
```

#### OUTPUT:-

A screenshot of a terminal window with a black background and yellow text. The output shows five lines of task execution messages followed by a concluding statement.

```
Executing Task One  
Executing Task Two  
Executing Task Three  
Executing Final Task  
This is a concrete class method.
```

**Q19. Write a Java program to show the use of all keywords for exception handling.**

```
package assignment1;
```

```
public class Question19 {  
    public static void main(String[] args) {  
        try {  
            int[] numbers = new int[3];  
            numbers[5] = 50;  
        }  
        catch (ArrayIndexOutOfBoundsException ex) {  
            System.out.println("Caught ArrayIndexOutOfBoundsException.");  
        }  
  
        try {  
            int x = 5;  
            int y = 0;  
            int result = x / y;  
        }  
        catch (ArithmeticException ex) {  
            System.out.println("Caught ArithmeticException.");  
        }  
  
        try {  
            String text = null;  
            System.out.println(text.length());  
        }  
        catch (NullPointerException ex) {  
            System.out.println("Caught NullPointerException.");  
        }  
  
        try {  
            int[] numbers = new int[2];  
            numbers[3] = 10;  
        }  
        catch (ArrayIndexOutOfBoundsException ex) {  
            System.out.println("Caught an exception in the catch block.");  
        }  
        finally {  
            System.out.println("This is the finally block. It always runs.");  
        }  
    }  
}
```

```
try {  
    String value = "abc";  
    int num = Integer.parseInt(value);  
}  
catch (NumberFormatException ex) {  
    System.out.println("Caught NumberFormatException.");  
}  
catch (Exception ex) {  
    System.out.println("Caught a generic exception.");  
}  
}  
}
```

### OUTPUT:-

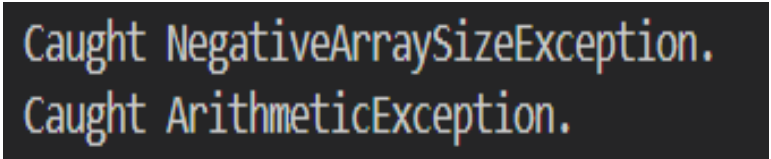
```
Caught ArrayIndexOutOfBoundsException.  
Caught ArithmeticException.  
Caught NullPointerException.  
Caught an exception in the catch block.  
This is the finally block. It always runs.  
Caught NumberFormatException.
```

**Q20. Write a Java program using try and catch to generate NegativeArrayIndex Exception and Arithmetic Exception.**

```
package assignment1;
```

```
public class Question20 {  
    public static void main(String[] args) {  
        try {  
            int[] numbers = new int[-3];  
        }  
        catch (NegativeArraySizeException ex) {  
            System.out.println("Caught NegativeArraySizeException.");  
        }  
  
        try {  
            int a = 10;  
            int b = 0;  
            int result = a / b;  
        }  
        catch (ArithmeticException ex) {  
            System.out.println("Caught ArithmeticException.");  
        }  
    }  
}
```

**OUTPUT:-**



```
Caught NegativeArraySizeException.  
Caught ArithmeticException.
```

**Q21. Write a program that outputs the name of the capital of the country entered at the command line. The program should throw a “NoMatchFoundException” when it fails to print the capital of the country entered at the command line..**

```
package assignment1;
import java.util.Scanner;

public class Question21 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter a country name:");
        String country = scanner.nextLine();

        try {
            String capital = findCapital(country);
            System.out.println("The capital of " + country + " is " + capital);
        }
        catch (NoMatchFoundException e) {
            System.out.println(e.getMessage());
        }

        scanner.close();
    }

    public static String findCapital(String country) throws NoMatchFoundException {
        if (country.equalsIgnoreCase("India")) {
            return "New Delhi";
        }
        else if (country.equalsIgnoreCase("USA")) {
            return "Washington D.C.";
        }
        else if (country.equalsIgnoreCase("Japan")) {
            return "Tokyo";
        }
        else if (country.equalsIgnoreCase("Australia")) {
            return "Canberra";
        }
        else {
            throw new NoMatchFoundException("Sorry, no capital found for " + country + ".");
        }
    }
}
```

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

### OUTPUT:-

```
Enter a country name:  
Japan  
The capital of Japan is Tokyo
```

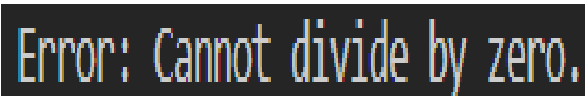
```
Enter a country name:  
Singapore  
Sorry, no capital found for Singapore.
```

**Q22. Write a java program to create an custom Exception that would handle at least 2 kind of Arithmetic Exceptions while calculating a given equation.**

```
package assignment1;
```

```
public class Question22 {  
    public static void main(String[] args) {  
        try {  
            int result1 = performOperation(10, 0);  
            int result2 = performOperation(10, 5);  
        } catch (MyArithmeticException e) {  
            System.out.println(e.getMessage());  
        }  
    }  
  
    public static int performOperation(int num1, int num2) throws MyArithmeticException {  
        if (num2 == 0) {  
            throw new MyArithmeticException("Error: Cannot divide by zero.");  
        }  
  
        if (num1 % num2 != 0) {  
            throw new MyArithmeticException("Error: The division is not an integer.");  
        }  
  
        return num1 / num2;  
    }  
}  
  
class MyArithmeticException extends Exception {  
    public MyArithmeticException(String message) {  
        super(message);  
    }  
}
```

**OUTPUT:-**

The output is displayed in a dark-themed terminal window. The text "Error: Cannot divide by zero." is shown in a light blue or cyan monospaced font, indicating an exception has been caught and printed.



**Q23. Create two user-defined exceptions named “TooHot” and “TooCold” to check the temperature (in Celsius) given by the user passed through the command line is too hot or too cold.**

**If temperature > 35, throw exception “TooHot”.**

**If temperature <5, throw exception “TooCold”.**

**Otherwise, print “Normal” and convert it to Farenheit.**

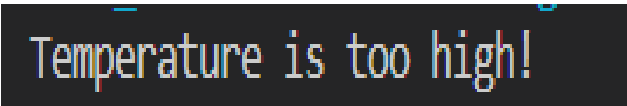
```
package assignment1;
```

```
public class Question23 {  
    public static void main(String[] args) {  
        int temp = 40;  
  
        try {  
            if (temp > 35) {  
                throw new HotTemperatureException("Temperature is too high!");  
            } else if (temp < 5) {  
                throw new ColdTemperatureException("Temperature is too low!");  
            } else {  
                System.out.println("Temperature is normal.");  
                double fahrenheit = temp * 9.0 / 5 + 32;  
                System.out.println("In Fahrenheit: " + fahrenheit);  
            }  
        } catch (HotTemperatureException | ColdTemperatureException e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

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

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

**OUTPUT:-**



```
Temperature is too high!
```

**Q24. Consider an Employee recruitment system that prints the candidate name based on the age criteria. The name and age of the candidate are taken as Input. Create two user-defined exceptions named "TooOlder" and "TooYounger".**

**If age > 45, throw exception "TooOlder".**

**If age < 20, throw exception "TooYounger".**

**Otherwise, print "Eligible" and print the name of the candidate.**


```
package assignment1;
```

```
public class Question24 {  
    public static void main(String[] args) {  
        int candidateAge = 47;  
        String candidateName = "Alice";  
  
        try {  
            if (candidateAge > 45) {  
                throw new OverAgeException("Candidate is too old to apply.");  
            } else if (candidateAge < 20) {  
                throw new UnderAgeException("Candidate is too young to apply.");  
            } else {  
                System.out.println(candidateName + " is eligible for recruitment.");  
            }  
        } catch (OverAgeException | UnderAgeException e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

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

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

**OUTPUT:-**



```
Candidate is too old to apply.
```

**Q25. Write a program to raise a user defined exception if username is less than 6 characters and password does not match.**

```
package assignment1;
```

```
public class Question25 {
    public static void main(String[] args) {
        String user = "john";
        String pass = "hello123";
        String confirmPass = "hello124";

        try {
            if (user.length() < 6) {
                throw new ShortUsernameException("Username must be at least 6 characters.");
            }
            if (!pass.equals(confirmPass)) {
                throw new PasswordMismatch("The passwords do not match.");
            }
        } catch (ShortUsernameException | PasswordMismatch e) {
            System.out.println(e.getMessage());
        }
    }
}

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

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

**OUTPUT:-**

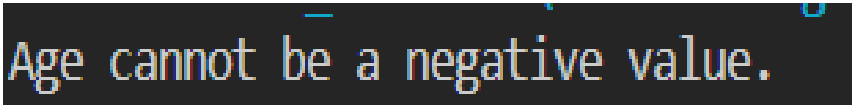
```
Username must be at least 6 characters.
```

**Q26. Write a program to input name and age of a person and throw a user-defined exception, if the entered age is negative.**

```
package assignment1;
```

```
public class Question26 {  
    public static void main(String[] args) {  
        String personName = "Alice";  
        int personAge = -2;  
  
        try {  
            if (personAge < 0) {  
                throw new InvalidAgeException("Age cannot be a negative value.");  
            } else {  
                System.out.println("The person " + personName + " is " + personAge + " years old.");  
            }  
        } catch (InvalidAgeException ex) {  
            System.out.println(ex.getMessage());  
        }  
    }  
}  
  
class InvalidAgeException extends Exception {  
    public InvalidAgeException(String msg) {  
        super(msg);  
    }  
}
```

**OUTPUT:-**



```
Age cannot be a negative value.
```