



Object Oriented Programming Using Java

PRACTICAL FILE (CSL – 344)

SUBMITTED TO :

DR. Pankaj Deep Kaur

SUBMITTED BY :

ABHINAV THAKUR

BTECH CSE 6TH SEM

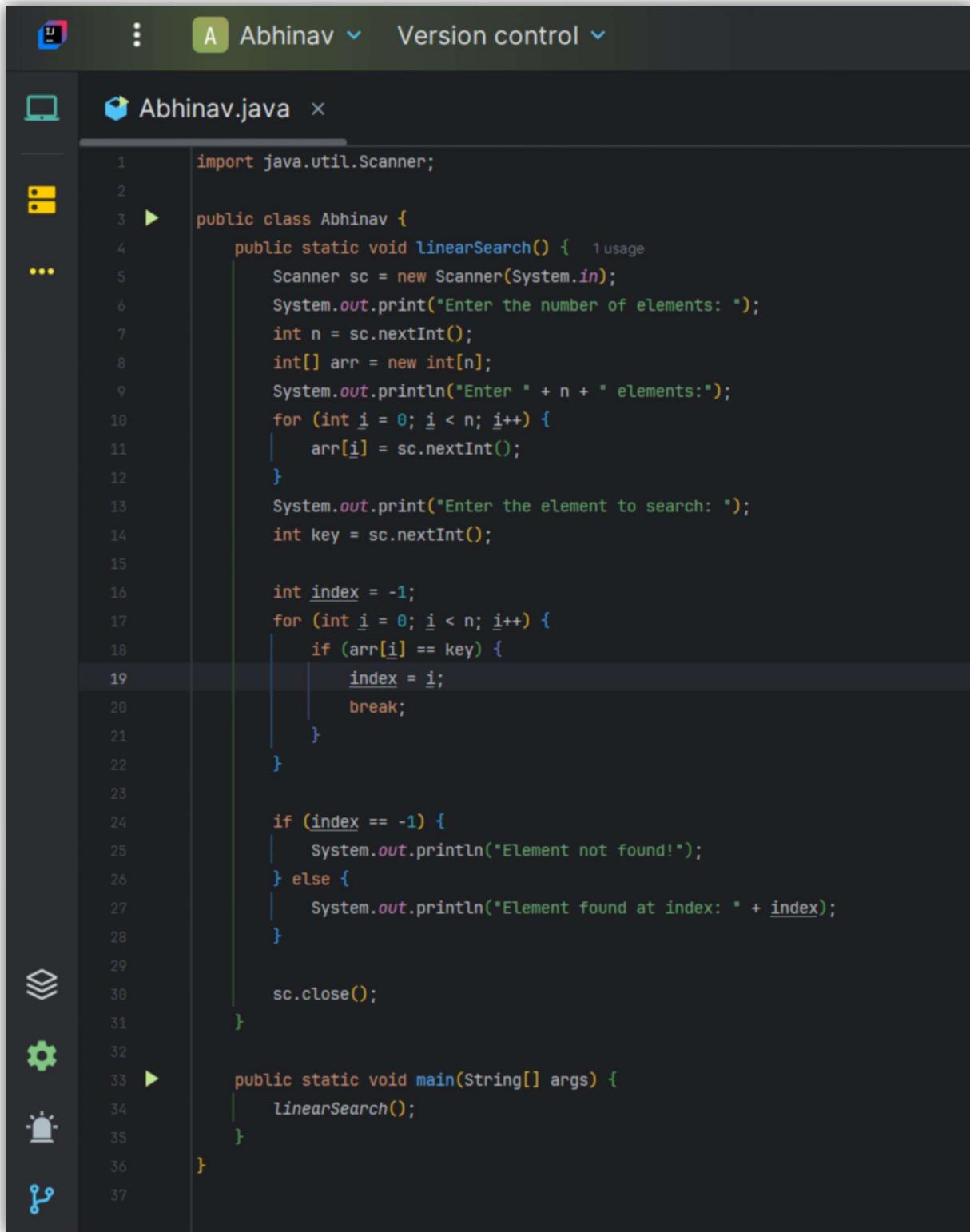
17032249064

INDEX

SR.NO.	TOPIC	Date	Remarks
1.	Write a Program to perform Linear search Algorithm.	22.01.2025	
2.	Write a Program to print Hello World.	22.01.2025	
3.	Write a Program to perform Binary search Algorithm	22.01.2025	
4.	Write a Program to perform Sorting Algorithms.(Bubble sort , Quick sort)	22.01.2025	
5.	Write a Program to find sum of two numbers using Command line arguments.	30-01-2025	
6.	Write a Program to show Arithmetic Operations (Addition , Subtraction , Multiplication and Divide)	30-01-2025	
7.	Write a Program to compute the following (i)-15+58*45 (ii)(35+8)%6 (iii)24+ -5*3/7 (iv)15+18/3*2-9%3	30-01-2025	
8.	Write a Program to compute the Area of (i)Circle (ii)Rectangle (iii)Triangle (iv)Square	30-01-2025	
9.	Write a Program to convert temperature from Fahrenheit to Degree Celcius	30-01-2025	
10.	Write a Program to sum values of single dimensional array.	03-02-2025	
11.	Write a Program to find average of elements of array	03-02-2025	
12.	Write a Program to remove a specific element form array	03-02-2025	
13.	Write a Program to insert an element (on a specific position) into a Multidimensional array.	03-02-2025	
14.	Write a program to perform the following operations on strings: 1. Compare two strings. 2. Count string length. 3. Convert upper case to lower case & vice versa. 4. Concatenate two strings. 5. Print a substring.	03-02-2025	

15.	Write a Program to show use of Scanner Class	03-02-2025	
16.	Write a Program to calculate the Area with the help of methods and object .	03-02-2025	
17.	Write a program to print Rectangular star Pattern.	03-02-2025	
18.	Write a program to print Right-angled triangle Pattern.	03-02-2025	
19.	Write a program to print Right-angled number Pyramid Pattern.	03-02-2025	
20.	Write a program to print Inverted Right Pyramid Pattern.	03-02-2025	
21.	Write a program to print Inverted Number Pyramid Pattern	03-02-2025	
22.	Write a program to print star Pyramid Pattern.	03-02-2025	
23.	Write a program to print Inverted star Pyramid Pattern.	03-02-2025	
24.	Write a program to print Diamond star Pattern.	03-02-2025	
25.	Write a program to print Half Diamond star Pattern.	03-02-2025	
26.	Write a program to print Binary Number Triangle Pattern.	03-02-2025	
27	Write a program to print Number crown Pattern.	03-02-2025	
28.	Write a program to print Increasing Number Triangle Pattern.	03-02-2025	
29.	Compute the average of three numbers through a Java Program.	03-02-2025	
30.	Construct a program to design a package in Java.	03-02-2025	
31.	Write a Java program that demonstrates the use of keywords: a) this b) static c) abstract	13-02-2025	
32.	WAP that demonstrates the use of constructor overloading concepts.	13-02-2025	
33.	WAP that exhibits the method overriding and method overloading.	13-02-2025	

1. Write a Program to perform Linear search Algorithm.



The screenshot shows a Java code editor interface with a dark theme. The top bar includes icons for file operations, a user profile labeled "Abhinav", and a "Version control" dropdown. The left sidebar features navigation icons for file, folder, and search functions. The main workspace displays the code for a Java class named "Abhinav".

```
import java.util.Scanner;

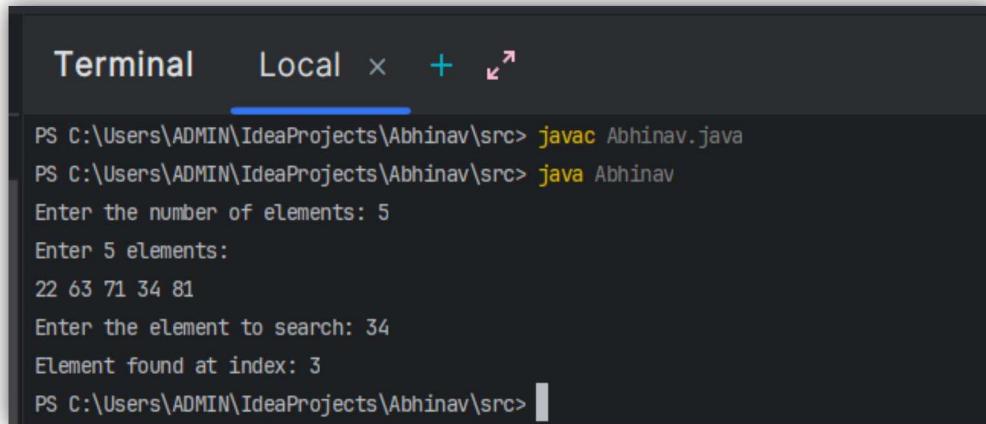
public class Abhinav {
    public static void linearSearch() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements: ");
        int n = sc.nextInt();
        int[] arr = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.print("Enter the element to search: ");
        int key = sc.nextInt();

        int index = -1;
        for (int i = 0; i < n; i++) {
            if (arr[i] == key) {
                index = i;
                break;
            }
        }

        if (index == -1) {
            System.out.println("Element not found!");
        } else {
            System.out.println("Element found at index: " + index);
        }
        sc.close();
    }

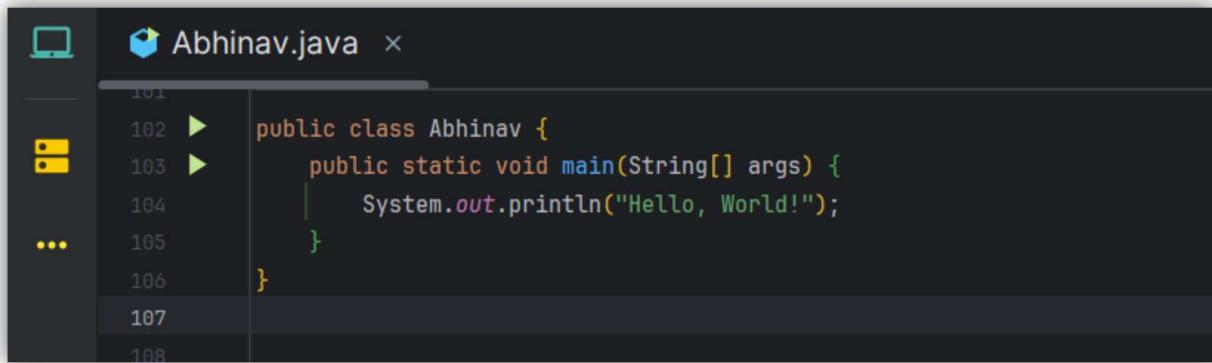
    public static void main(String[] args) {
        linearSearch();
    }
}
```

Output-



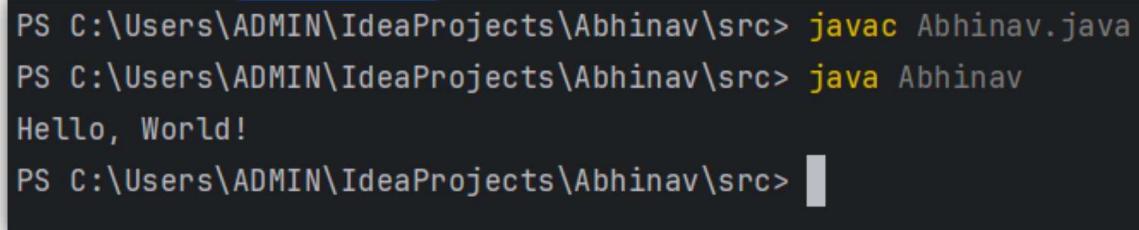
```
Terminal Local × + ↗
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter the number of elements: 5
Enter 5 elements:
22 63 71 34 81
Enter the element to search: 34
Element found at index: 3
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

2. Write a Program to print Hello World



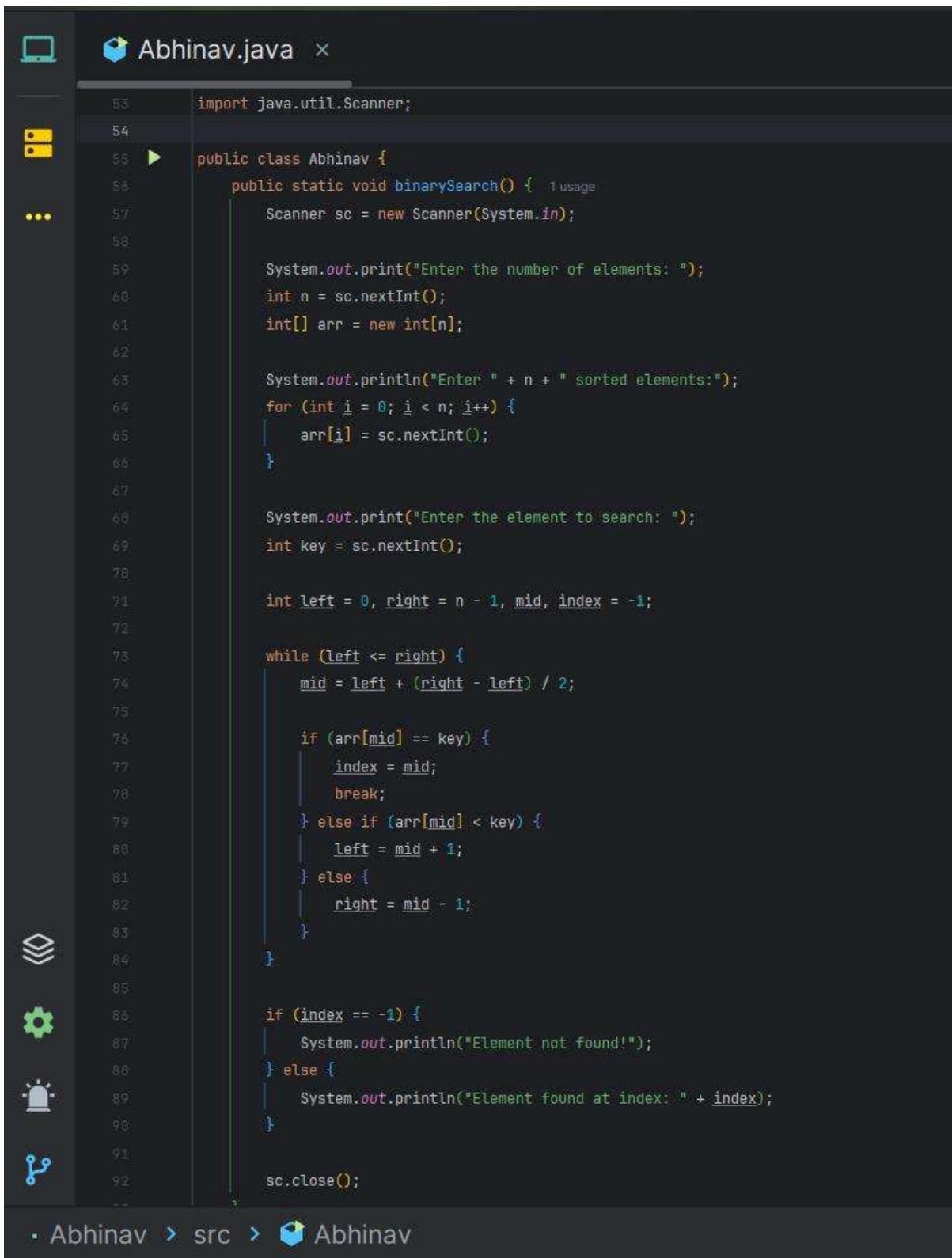
```
Abhinav.java ×
101
102 ► public class Abhinav {
103 ►   public static void main(String[] args) {
104     System.out.println("Hello, World!");
105   }
106 }
107
108
```

Output –



```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Hello, World!
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

3. Write a Program to perform Binary search Algorithm.



The screenshot shows a Java code editor with a dark theme. The file being edited is named `Abhinav.java`. The code implements a binary search algorithm. It starts by importing the `java.util.Scanner` class. The `binarySearch` method prompts the user to enter the number of elements and the elements themselves. It then asks for the element to search for. The algorithm uses a while loop to find the element, calculating the mid-point of the current search range and comparing it with the target value. If found, it prints the index; if not found, it prints a message indicating the element was not found. Finally, it closes the scanner.

```
import java.util.Scanner;

public class Abhinav {
    public static void binarySearch() {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the 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 the element to search: ");
        int key = sc.nextInt();

        int left = 0, right = n - 1, mid, index = -1;

        while (left <= right) {
            mid = left + (right - left) / 2;

            if (arr[mid] == key) {
                index = mid;
                break;
            } else if (arr[mid] < key) {
                left = mid + 1;
            } else {
                right = mid - 1;
            }
        }

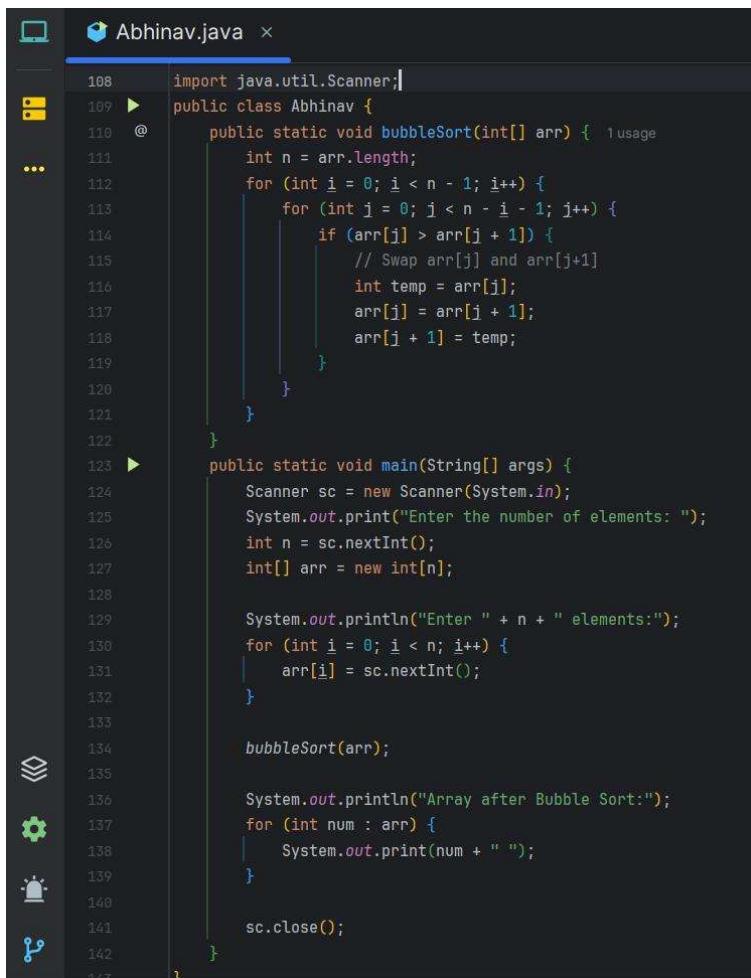
        if (index == -1) {
            System.out.println("Element not found!");
        } else {
            System.out.println("Element found at index: " + index);
        }
    }
}
```

• Abhinav > src > Abhinav

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter the number of elements: 5
Enter 5 sorted elements:
22 33 44 55 66
Enter the element to search: 33
Element found at index: 1
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

4. Write a Program to perform Sorting Algorithms.(Bubble sort , Quick sort)



The screenshot shows a Java code editor with the file 'Abhinav.java' open. The code contains two sorting methods: bubbleSort and main. The bubbleSort method uses nested loops to compare adjacent elements and swap them if they are in the wrong order. The main method prompts the user for the number of elements and the elements themselves, then calls bubbleSort and prints the sorted array.

```
108 import java.util.Scanner;
109 public class Abhinav {
110     @
111         public static void bubbleSort(int[] arr) {
112             int n = arr.length;
113             for (int i = 0; i < n - 1; i++) {
114                 for (int j = 0; j < n - i - 1; j++) {
115                     if (arr[j] > arr[j + 1]) {
116                         // Swap arr[j] and arr[j+1]
117                         int temp = arr[j];
118                         arr[j] = arr[j + 1];
119                         arr[j + 1] = temp;
120                     }
121                 }
122             }
123         }
124         public static void main(String[] args) {
125             Scanner sc = new Scanner(System.in);
126             System.out.print("Enter the number of elements: ");
127             int n = sc.nextInt();
128             int[] arr = new int[n];
129
130             System.out.println("Enter " + n + " elements:");
131             for (int i = 0; i < n; i++) {
132                 arr[i] = sc.nextInt();
133             }
134
135             bubbleSort(arr);
136
137             System.out.println("Array after Bubble Sort:");
138             for (int num : arr) {
139                 System.out.print(num + " ");
140             }
141
142             sc.close();
143         }
144 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter the number of elements: 5
Enter 5 elements:
50 10 31 20 40
Array after Bubble Sort:
10 20 31 40 50
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

The screenshot shows a Java code editor with a dark theme. The file is named `Abhinav.java`. The code implements a quicksort algorithm. A yellow lightbulb icon is placed next to the line `int n = sc.nextInt();`, indicating a potential issue or suggestion. The code is as follows:

```
import java.util.Scanner;

public class Abhinav {
    static void quickSort(int[] arr, int low, int high) {
        if (low < high) {
            int pivot = partition(arr, low, high);
            quickSort(arr, low, high - 1);
            quickSort(arr, low + 1, high);
        }
    }

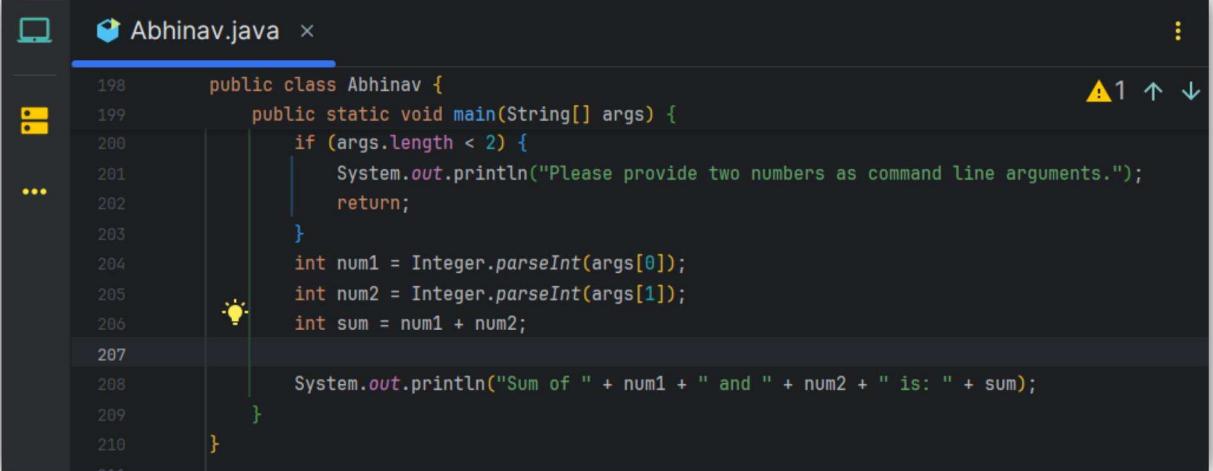
    static int partition(int[] arr, int low, int high) {
        int pivot = arr[high], i = low - 1;
        for (int j = low; j < high; j++) {
            if (arr[j] < pivot) {
                int temp = arr[++i]; arr[i] = arr[j]; arr[j] = temp;
            }
        }
        int temp = arr[i + 1]; arr[i + 1] = arr[high]; arr[high] = temp;
        return i + 1;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of elements: ");
        int n = sc.nextInt(), arr[] = new int[n];
        System.out.println("Enter " + n + " elements:");
        for (int i = 0; i < n; arr[i++] = sc.nextInt());
        quickSort(arr, low: 0, high: n - 1);
        System.out.println("Sorted Array:");
        for (int num : arr) System.out.print(num + " ");
        sc.close();
    }
}
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of elements: 5
Enter 5 elements:
20 12 48 29 83
Sorted Array:
12 20 29 48 83
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

5. Write a Program to find sum of two numbers using Command line arguments.

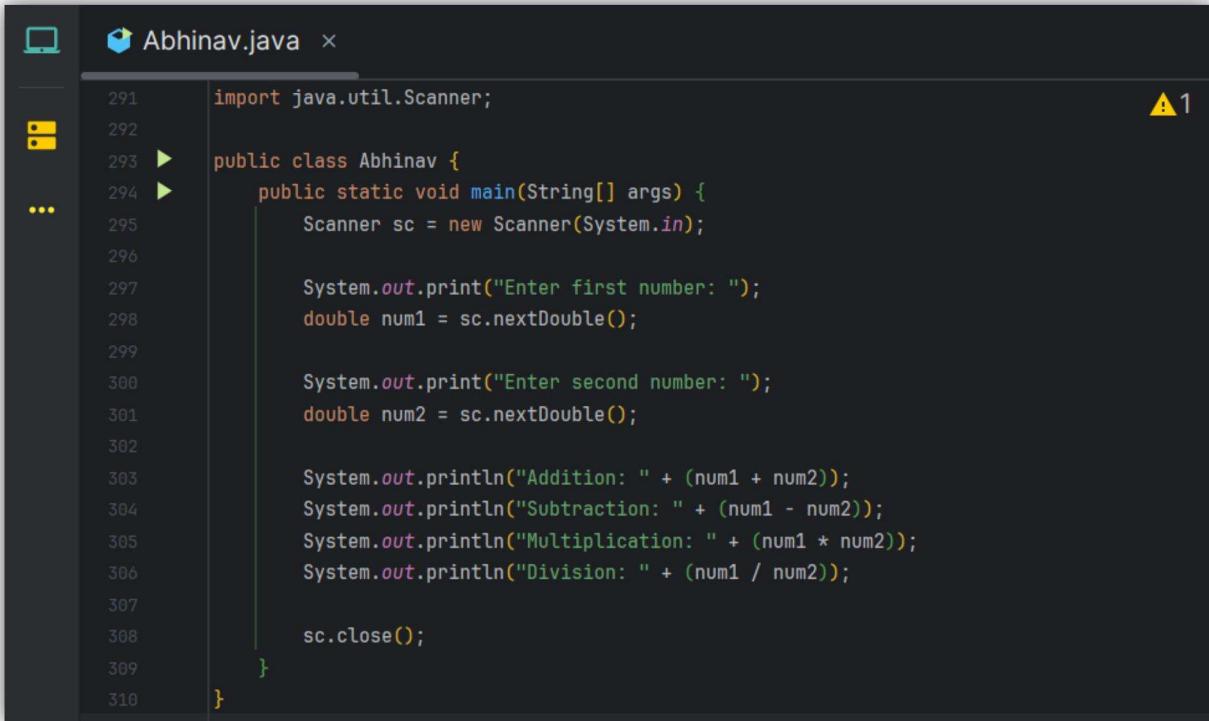


```
Abhinav.java
198 public class Abhinav {
199     public static void main(String[] args) {
200         if (args.length < 2) {
201             System.out.println("Please provide two numbers as command line arguments.");
202             return;
203         }
204         int num1 = Integer.parseInt(args[0]);
205         int num2 = Integer.parseInt(args[1]);
206         int sum = num1 + num2;
207
208         System.out.println("Sum of " + num1 + " and " + num2 + " is: " + sum);
209     }
210 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav 20 30
Sum of 20 and 30 is: 50
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

6. Write a Program to show Arithmetic Operations (Addition, Subtraction , Multiplication and Divide)



The screenshot shows a Java code editor with a dark theme. The file is named 'Abhinav.java'. The code implements a simple program that reads two double numbers from the user and prints their sum, difference, product, and quotient. A yellow warning icon in the top right corner indicates one unresolved reference.

```
import java.util.Scanner;
public class Abhinav {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first number: ");
        double num1 = sc.nextDouble();
        System.out.print("Enter second number: ");
        double num2 = sc.nextDouble();
        System.out.println("Addition: " + (num1 + num2));
        System.out.println("Subtraction: " + (num1 - num2));
        System.out.println("Multiplication: " + (num1 * num2));
        System.out.println("Division: " + (num1 / num2));
        sc.close();
    }
}
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter first number: 50
Enter second number: 10
Addition: 60.0
Subtraction: 40.0
Multiplication: 500.0
Division: 5.0
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

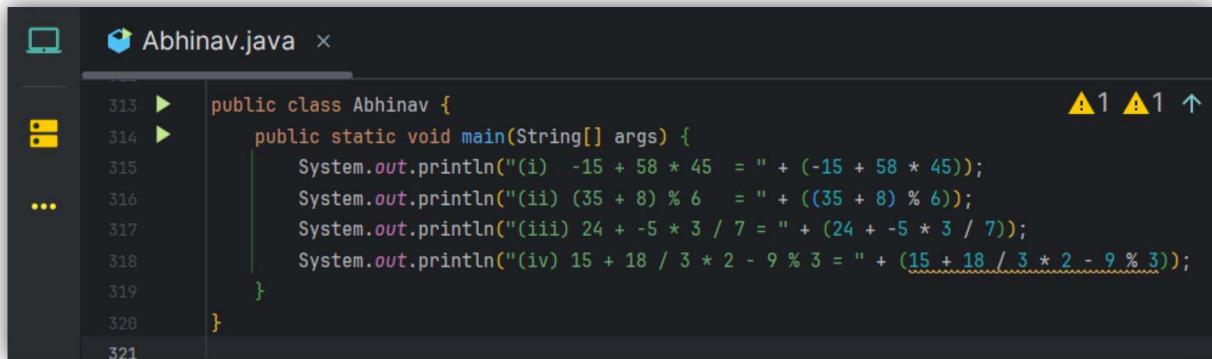
7. Write a Program to compute the following

(i) $-15 + 58 * 45$

(ii) $(35 + 8) \% 6$

(iii) $24 + -5 * 3 / 7$

(iv) $15 + 18 / 3 * 2 - 9 \% 3$



The screenshot shows a Java code editor with a file named "Abhinav.java". The code defines a class "Abhinav" with a main method that prints four results using System.out.println. The results correspond to the expressions listed above. The code is as follows:

```
public class Abhinav {
    public static void main(String[] args) {
        System.out.println("(i) -15 + 58 * 45 = " + (-15 + 58 * 45));
        System.out.println("(ii) (35 + 8) % 6 = " + ((35 + 8) % 6));
        System.out.println("(iii) 24 + -5 * 3 / 7 = " + (24 + -5 * 3 / 7));
        System.out.println("(iv) 15 + 18 / 3 * 2 - 9 % 3 = " + (15 + 18 / 3 * 2 - 9 % 3));
    }
}
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
(i) -15 + 58 * 45 = 2595
(ii) (35 + 8) % 6 = 1
(iii) 24 + -5 * 3 / 7 = 22
(iv) 15 + 18 / 3 * 2 - 9 % 3 = 27
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

8. Write a Program to compute the Area of (i)Circle (ii)Rectangle (iii)Triangle (iv)Square

The screenshot shows a Java code editor with a dark theme. The file is named `Abhinav.java`. The code is a single class with a main method that calculates the area of four different shapes: Circle, Rectangle, Triangle, and Square. The code uses `System.out.print` and `System.out.println` to interact with the user for input and output.

```
public class Abhinav {
    public static void main(String[] args) {
        // Circle
        System.out.print("Enter radius of circle: ");
        double radius = sc.nextDouble();
        double circleArea = Math.PI * radius * radius;
        System.out.println("Area of Circle: " + circleArea);

        // Rectangle
        System.out.print("Enter length of rectangle: ");
        double length = sc.nextDouble();
        System.out.print("Enter width of rectangle: ");
        double width = sc.nextDouble();
        double rectangleArea = length * width;
        System.out.println("Area of Rectangle: " + rectangleArea);

        // Triangle
        System.out.print("Enter base of triangle: ");
        double base = sc.nextDouble();
        System.out.print("Enter height of triangle: ");
        double height = sc.nextDouble();
        double triangleArea = 0.5 * base * height;
        System.out.println("Area of Triangle: " + triangleArea);

        // Square
        System.out.print("Enter side of square: ");
        double side = sc.nextDouble();
        double squareArea = side * side;
        System.out.println("Area of Square: " + squareArea);

        sc.close();
    }
}
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter radius of circle: 5
Area of Circle: 78.53981633974483
Enter length of rectangle: 10
Enter width of rectangle: 4
Area of Rectangle: 40.0
Enter base of triangle: 8
Enter height of triangle: 6
Area of Triangle: 24.0
Enter side of square: 7
Area of Square: 49.0
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

9. Write a Program to convert temperature from Fahrenheit to Degree Celcius.



The screenshot shows a Java code editor with a file named "Abhinav.java". The code is as follows:

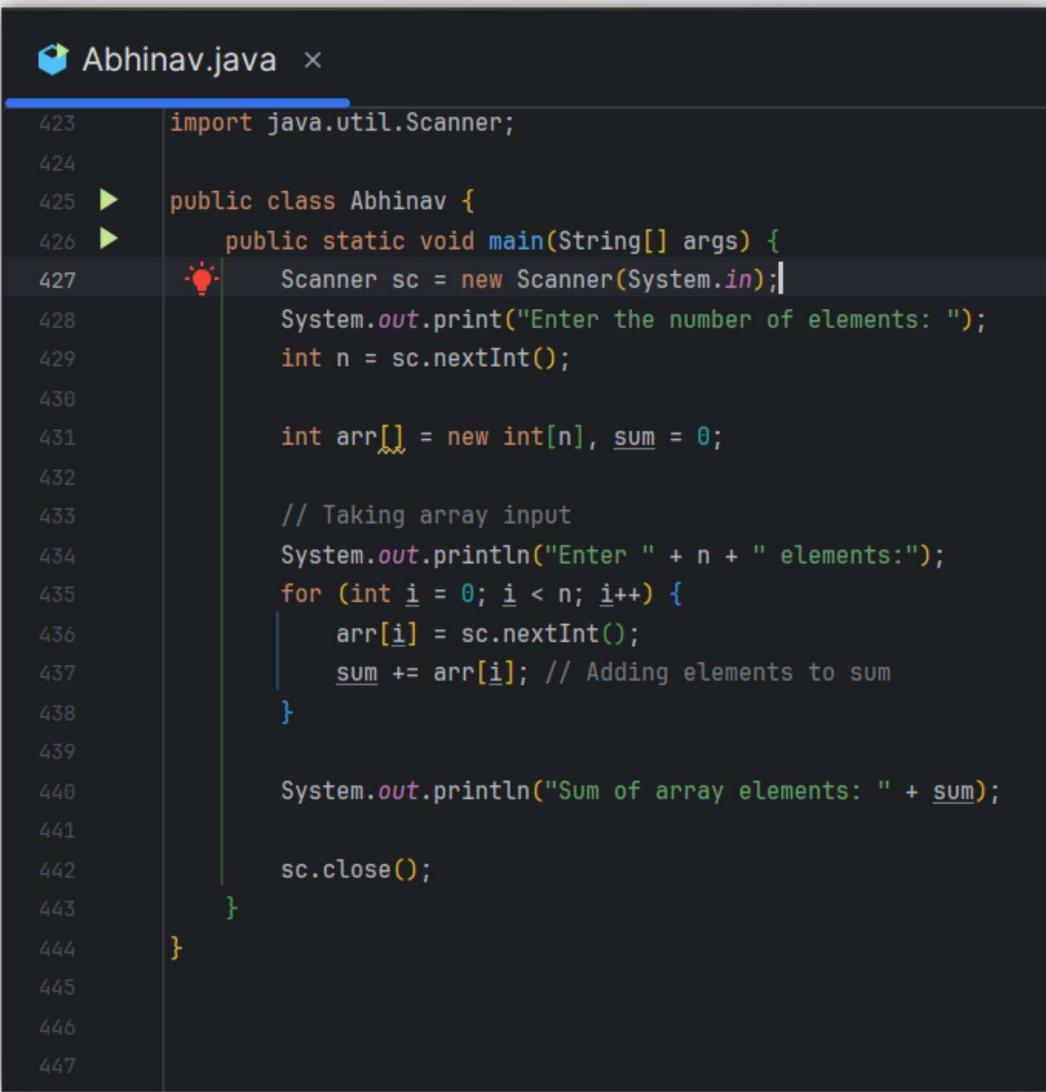
```
404
405     import java.util.Scanner;
406
407 ►  public class Abhinav {
408 ►      public static void main(String[] args) {
409         Scanner sc = new Scanner(System.in);
410         System.out.print("Enter temperature in Fahrenheit: ");
411         double fahrenheit = sc.nextDouble();
412
413         // Convert to Celsius
414         double celsius = (fahrenheit - 32) * 5 / 9;
415         System.out.println("Temperature in Celsius: " + celsius);
416
417         sc.close();
418     }
419 }
```

A yellow warning icon with the number "1" is visible in the top right corner of the code editor window.

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter temperature in Fahrenheit: 101
Temperature in Celsius: 38.33333333333336
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

10. Write a Program to sum values of single dimensional array.



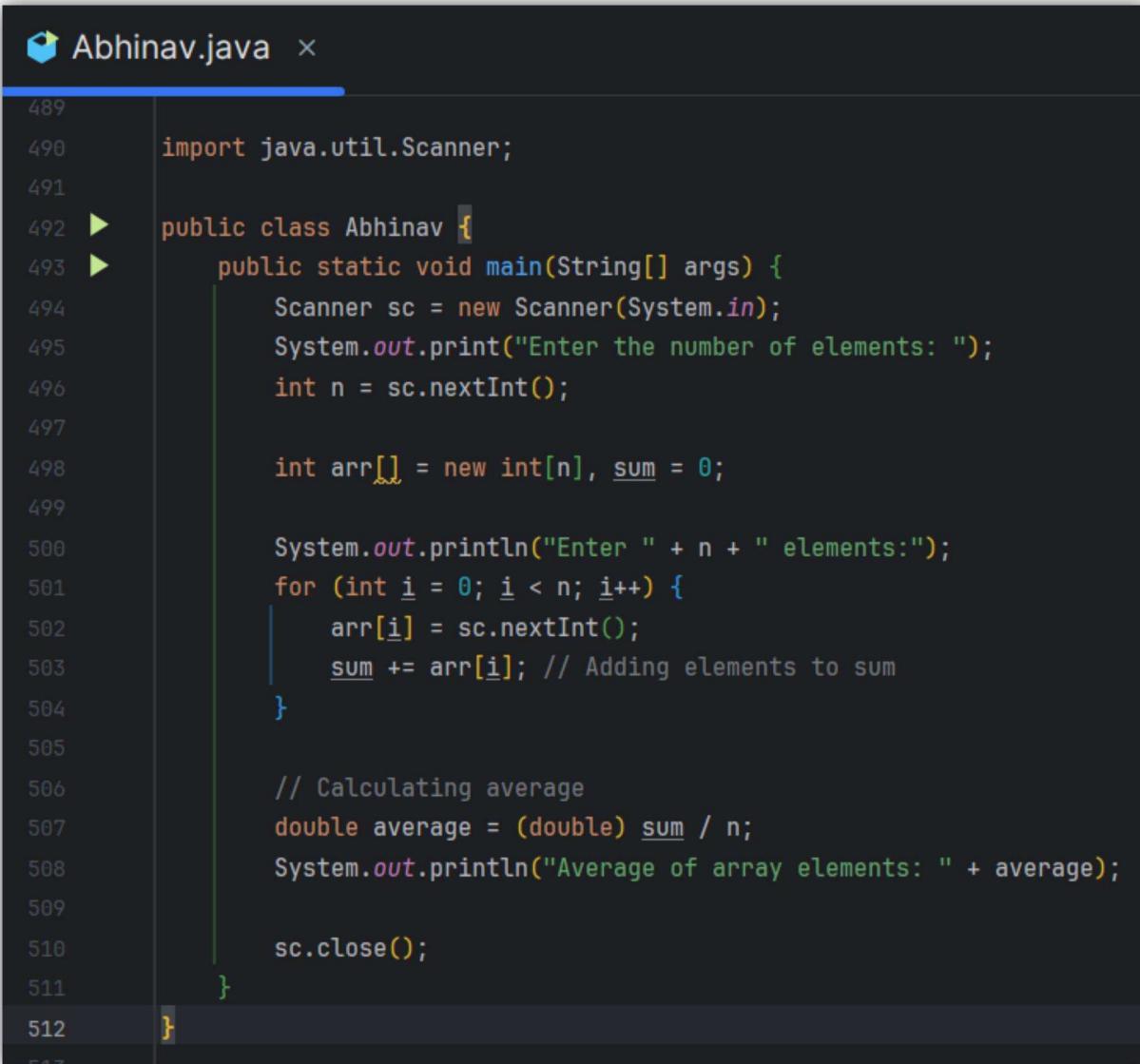
The screenshot shows a code editor window for a Java file named "Abhinav.java". The code is a simple program that prompts the user for the number of elements in an array, takes input for each element, calculates the sum of all elements, and prints the result. The code uses the Scanner class to handle user input and System.out.println to output results. The code editor interface includes line numbers on the left, syntax highlighting, and a status bar at the bottom.

```
423 import java.util.Scanner;
424
425 public class Abhinav {
426     public static void main(String[] args) {
427         Scanner sc = new Scanner(System.in);
428         System.out.print("Enter the number of elements: ");
429         int n = sc.nextInt();
430
431         int arr[] = new int[n], sum = 0;
432
433         // Taking array input
434         System.out.println("Enter " + n + " elements:");
435         for (int i = 0; i < n; i++) {
436             arr[i] = sc.nextInt();
437             sum += arr[i]; // Adding elements to sum
438         }
439
440         System.out.println("Sum of array elements: " + sum);
441
442         sc.close();
443     }
444 }
445
446
447
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter the number of elements: 5
Enter 5 elements:
15 25 35 45 55
Sum of array elements: 175
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

11. Write a Program to find average of elements of array.



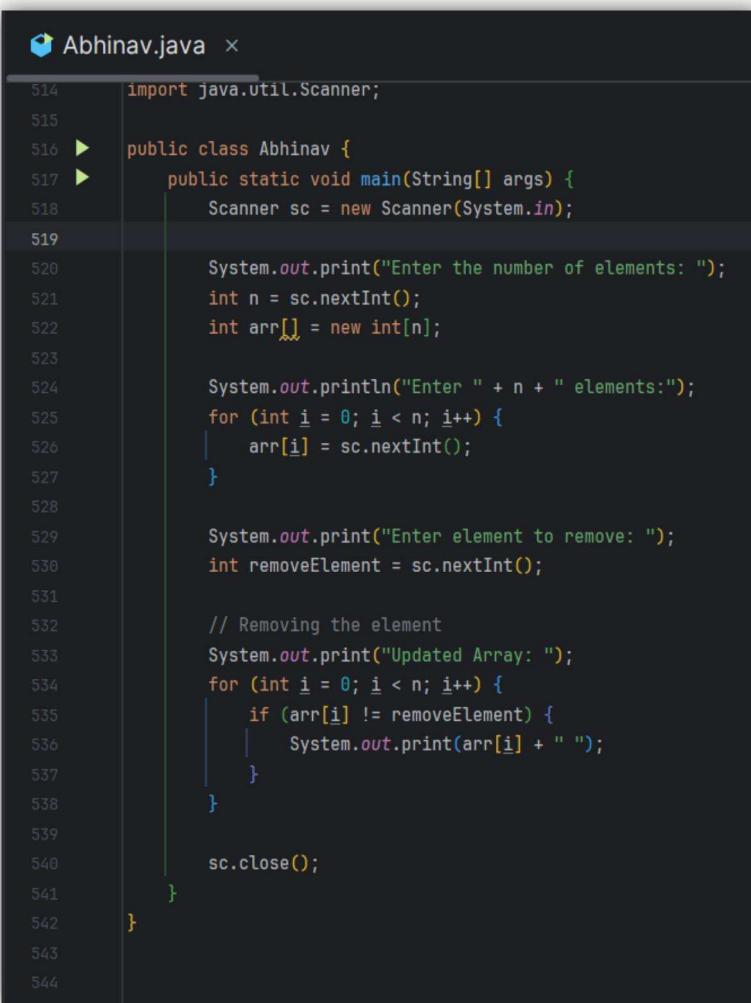
The screenshot shows a Java code editor with the file 'Abhinav.java' open. The code is a simple program that calculates the average of elements in an array. It uses a Scanner to input the number of elements and the elements themselves, then calculates the sum and divides it by the number of elements to find the average. The code is numbered from 489 to 512 on the left.

```
489
490     import java.util.Scanner;
491
492 ► public class Abhinav {
493 ►     public static void main(String[] args) {
494         Scanner sc = new Scanner(System.in);
495         System.out.print("Enter the number of elements: ");
496         int n = sc.nextInt();
497
498         int arr[] = new int[n], sum = 0;
499
500         System.out.println("Enter " + n + " elements:");
501         for (int i = 0; i < n; i++) {
502             arr[i] = sc.nextInt();
503             sum += arr[i]; // Adding elements to sum
504         }
505
506         // Calculating average
507         double average = (double) sum / n;
508         System.out.println("Average of array elements: " + average);
509
510         sc.close();
511     }
512 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter the number of elements: 6
Enter 6 elements:
12 24 36 48 60 72
Average of array elements: 42.0
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

12. Write a Program to remove a specific element from array.



The screenshot shows a Java code editor with the file 'Abhinav.java' open. The code is a program that prompts the user for the number of elements in an array, reads the elements into an array, and then removes a specified element from the array. The code uses a Scanner object to read input from the console.

```
514 import java.util.Scanner;
515
516 public class Abhinav {
517     public static void main(String[] args) {
518         Scanner sc = new Scanner(System.in);
519
520         System.out.print("Enter the number of elements: ");
521         int n = sc.nextInt();
522         int arr[] = new int[n];
523
524         System.out.println("Enter " + n + " elements:");
525         for (int i = 0; i < n; i++) {
526             arr[i] = sc.nextInt();
527         }
528
529         System.out.print("Enter element to remove: ");
530         int removeElement = sc.nextInt();
531
532         // Removing the element
533         System.out.print("Updated Array: ");
534         for (int i = 0; i < n; i++) {
535             if (arr[i] != removeElement) {
536                 System.out.print(arr[i] + " ");
537             }
538         }
539
540         sc.close();
541     }
542 }
543
544 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter the number of elements: 5
Enter 5 elements:
22 44 66 88 110
Enter element to remove: 88
Updated Array: 22 44 66 110
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

13. Write a Program to insert an element (on a specific position) into a Multidimensional array.

```
544 import java.util.Scanner;
545
546 public class Abhinav {
547     public static void main(String[] args) {
548         Scanner sc = new Scanner(System.in);
549
550         // Taking row and column size
551         System.out.print("Enter number of rows: ");
552         int rows = sc.nextInt();
553         System.out.print("Enter number of columns: ");
554         int cols = sc.nextInt();
555
556         int arr[][] = new int[rows][cols];
557         // Taking 2D array input
558         System.out.println("Enter " + (rows * cols) + " elements:");
559         for (int i = 0; i < rows; i++) {
560             for (int j = 0; j < cols; j++) {
561                 arr[i][j] = sc.nextInt();
562             }
563         }
564
565         // Taking position and new value
566         System.out.print("Enter row index to insert: ");
567         int rowIndex = sc.nextInt();
568         System.out.print("Enter column index to insert: ");
569         int colIndex = sc.nextInt();
570         System.out.print("Enter new value: ");
571         int newValue = sc.nextInt();
572
573         // Inserting new value
574         arr[rowIndex][colIndex] = newValue;
575         System.out.println("Updated 2D Array:");
576         for (int i = 0; i < rows; i++) {
577             for (int j = 0; j < cols; j++) {
578                 System.out.print(arr[i][j] + " ");
579             }
580             System.out.println();
581         }
582     }
583     sc.close();
584 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 2
Enter number of columns: 3
Enter 6 elements:
1 2 3
4 5 6
Enter row index to insert: 1
Enter column index to insert: 2
Enter new value: 55
Updated 2D Array:
1 2 3
4 5 55
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

14. Write a program to perform the following operations on strings:

- 1. Compare two strings.**
- 2. Count string length.**
- 3. Convert upper case to lower case & vice versa.**
- 4. Concatenate two strings.**
- 5. Print a substring.**

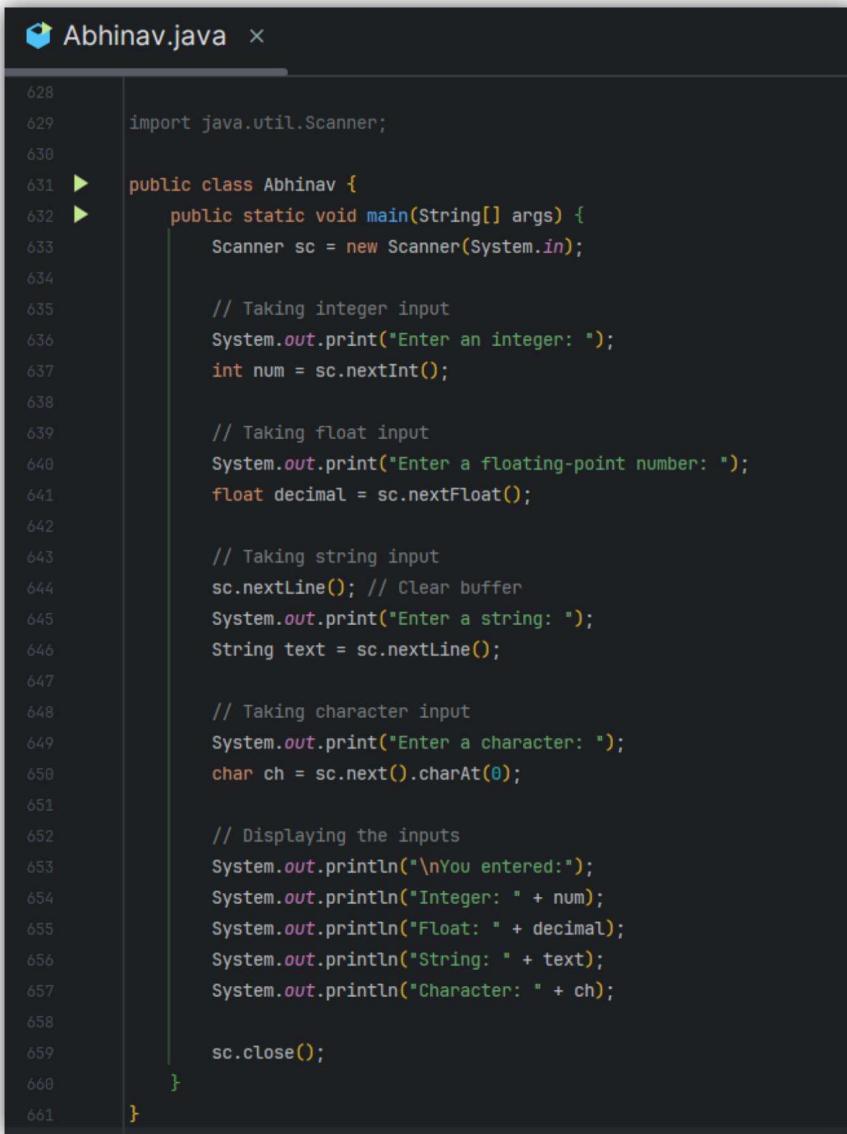
The screenshot shows a Java code editor with the file 'Abhinav.java' open. The code implements several string manipulation methods. It uses a Scanner to input two strings from the user. It compares them, counts their lengths, converts them to uppercase and lowercase, concatenates them, and prints substrings based on user-specified start and end indices. The code is annotated with line numbers from 587 to 624.

```
587
588 ► public class Abhinav {
589 ►   public static void main(String[] args) {
590     Scanner sc = new Scanner(System.in);
591
592     System.out.print("Enter first string: ");
593     String str1 = sc.nextLine();
594     System.out.print("Enter second string: ");
595     String str2 = sc.nextLine();
596     // 1. Compare two strings
597     if (str1.equals(str2)) {
598       System.out.println("Both strings are equal.");
599     } else {
600       System.out.println("Strings are not equal.");
601     }
602     // 2. Count string length
603     System.out.println("Length of first string: " + str1.length());
604     System.out.println("Length of second string: " + str2.length());
605
606     // 3. Convert case
607     System.out.println("Uppercase of first string: " + str1.toUpperCase());
608     System.out.println("Lowercase of first string: " + str1.toLowerCase());
609
610     // 4. Concatenate two strings
611     String concatenated = str1 + " " + str2;
612     System.out.println("Concatenated string: " + concatenated);
613
614     // 5. Print a substring
615     System.out.print("Enter start index for substring (0-based index): ");
616     int startIndex = sc.nextInt();
617     System.out.print("Enter end index for substring: ");
618     int endIndex = sc.nextInt();
619     if (startIndex >= 0 && endIndex <= str1.length() && startIndex < endIndex) {
620       System.out.println("Substring: " + str1.substring(startIndex, endIndex));
621     } else {
622       System.out.println("Invalid substring range!");
623     }
624     sc.close();
}
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter first string: Hello
Enter second string: World
Strings are not equal.
Length of first string: 5
Length of second string: 5
Uppercase of first string: HELLO
Lowercase of first string: hello
Concatenated string: Hello World
Enter start index for substring (0-based index): 1
Enter end index for substring: 4
Substring: ell
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

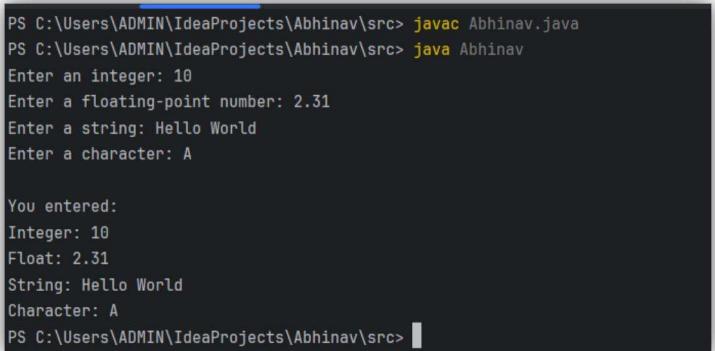
15. Write a Program to show use of Scanner Class.



The screenshot shows a code editor window with a dark theme. The file is named 'Abhinav.java'. The code demonstrates various methods of the Scanner class to read input from the standard input stream (System.in). It includes reading integers, floating-point numbers, strings, and characters, and then printing the entered values back to the standard output stream (System.out).

```
628
629     import java.util.Scanner;
630
631 ► public class Abhinav {
632 ►     public static void main(String[] args) {
633         Scanner sc = new Scanner(System.in);
634
635         // Taking integer input
636         System.out.print("Enter an integer: ");
637         int num = sc.nextInt();
638
639         // Taking float input
640         System.out.print("Enter a floating-point number: ");
641         float decimal = sc.nextFloat();
642
643         // Taking string input
644         sc.nextLine(); // Clear buffer
645         System.out.print("Enter a string: ");
646         String text = sc.nextLine();
647
648         // Taking character input
649         System.out.print("Enter a character: ");
650         char ch = sc.next().charAt(0);
651
652         // Displaying the inputs
653         System.out.println("\nYou entered:");
654         System.out.println("Integer: " + num);
655         System.out.println("Float: " + decimal);
656         System.out.println("String: " + text);
657         System.out.println("Character: " + ch);
658
659         sc.close();
660     }
661 }
```

Output –

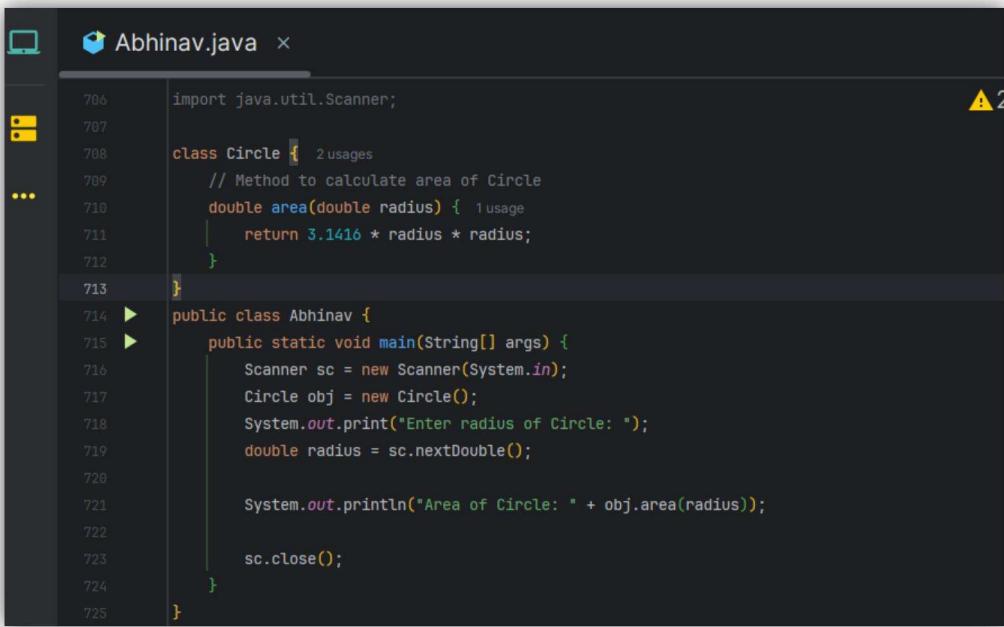


The terminal window shows the command to compile ('javac') and run ('java') the 'Abhinav' program. It then prompts for four types of input: an integer, a floating-point number, a string, and a character. Finally, it prints out each input followed by a descriptive message.

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter an integer: 10
Enter a floating-point number: 2.31
Enter a string: Hello World
Enter a character: A

You entered:
Integer: 10
Float: 2.31
String: Hello World
Character: A
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

16 . Write a Program to calculate the Area with the help of methods and object .



The screenshot shows a Java code editor with the file 'Abhinav.java' open. The code defines a class 'Circle' with a static method 'area' that calculates the area of a circle given its radius. It also defines a class 'Abhinav' with a main method that prompts the user for a radius, creates a 'Circle' object, and prints the calculated area. There are two yellow warning icons in the top right corner of the code editor.

```
706 import java.util.Scanner;
707
708 class Circle {
709     // Method to calculate area of Circle
710     double area(double radius) {
711         return 3.1416 * radius * radius;
712     }
713 }
714 public class Abhinav {
715     public static void main(String[] args) {
716         Scanner sc = new Scanner(System.in);
717         Circle obj = new Circle();
718         System.out.print("Enter radius of Circle: ");
719         double radius = sc.nextDouble();
720
721         System.out.println("Area of Circle: " + obj.area(radius));
722
723         sc.close();
724     }
725 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter radius of Circle: 5
Area of Circle: 78.54
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

17. Write a program to print Rectangular star Pattern.

The screenshot shows a Java code editor with a file named 'Abhinav.java'. The code prints a right-angled triangle pattern of asterisks based on user input for rows and columns. The code uses Scanner for input and System.out for output.

```
727 import java.util.Scanner;
728
729 public class Abhinav {
730     public static void main(String[] args) {
731         Scanner sc = new Scanner(System.in);
732
733         System.out.print("Enter number of rows: ");
734         int rows = sc.nextInt();
735         System.out.print("Enter number of columns: ");
736         int cols = sc.nextInt();
737
738         for (int i = 0; i < rows; i++) {
739             for (int j = 0; j < cols; j++) {
740                 System.out.print("* ");
741             }
742             System.out.println();
743         }
744
745         sc.close();
746     }
747 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 4
Enter number of columns: 8
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

18. Write a program to print Right-angled triangle Pattern.

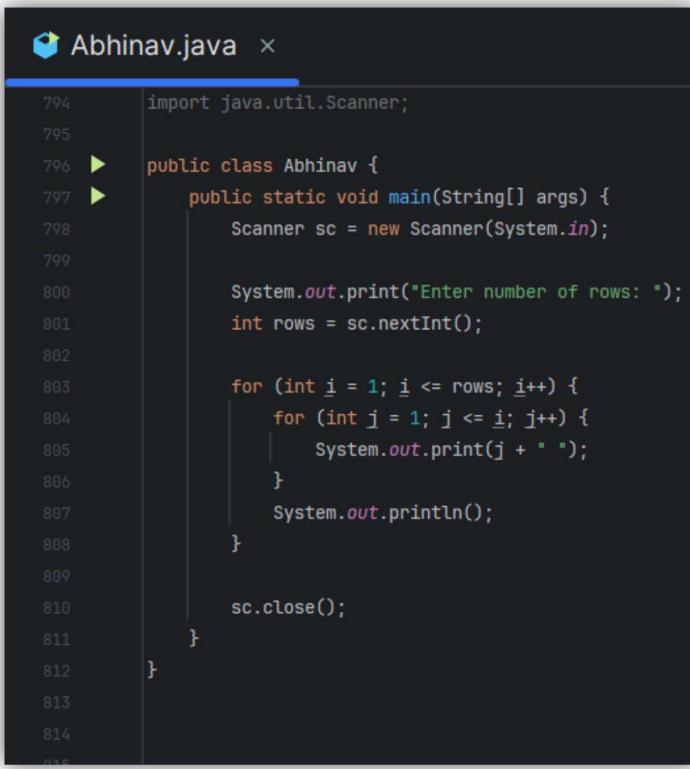
```
Abhinav.java ×

775 import java.util.Scanner;
776
777 ► public class Abhinav {
778 ►     public static void main(String[] args) {
779         Scanner sc = new Scanner(System.in);
780
781         System.out.print("Enter number of rows: ");
782         int rows = sc.nextInt();
783
784         for (int i = 1; i <= rows; i++) {
785             for (int j = 1; j <= i; j++) {
786                 System.out.print("* ");
787             }
788             System.out.println();
789         }
790
791         sc.close();
792     }
793 }
794
795
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 5
*
* *
* * *
* * * *
* * * * *
```

19. Write a program to print Right-angled number Pyramid Pattern.

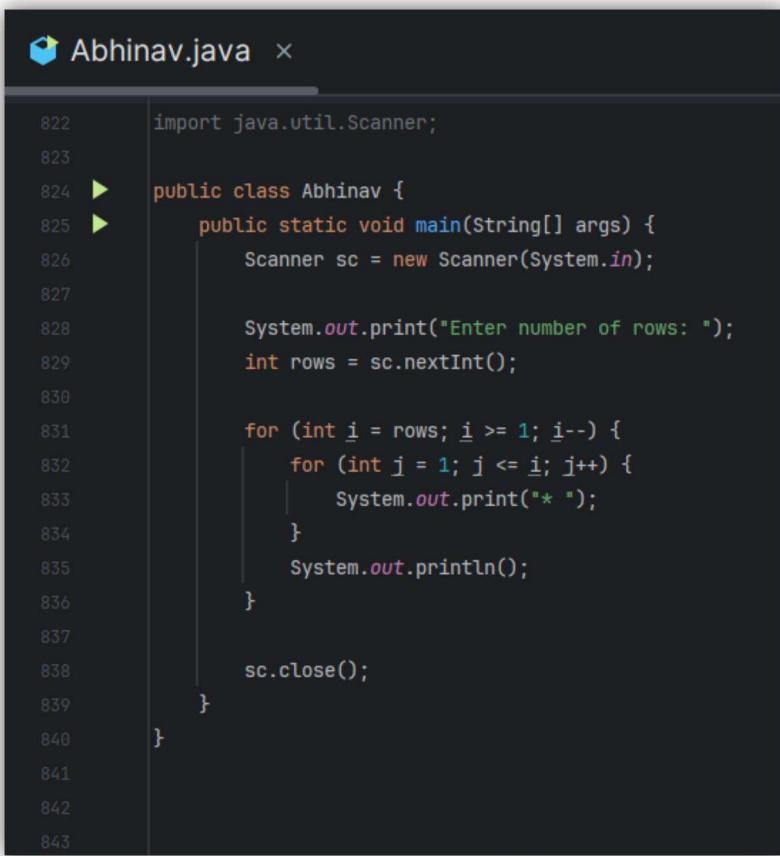


```
794 import java.util.Scanner;
795
796 public class Abhinav {
797     public static void main(String[] args) {
798         Scanner sc = new Scanner(System.in);
799
800         System.out.print("Enter number of rows: ");
801         int rows = sc.nextInt();
802
803         for (int i = 1; i <= rows; i++) {
804             for (int j = 1; j <= i; j++) {
805                 System.out.print(j + " ");
806             }
807             System.out.println();
808         }
809
810         sc.close();
811     }
812 }
813
814
815
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

20. Write a program to print Inverted Right Pyramid Pattern.



The screenshot shows a code editor window titled "Abhinav.java". The code is a Java program that prints an inverted pyramid pattern of asterisks. It uses a Scanner to read the number of rows from the user and then prints the pattern using nested loops.

```
822     import java.util.Scanner;
823
824 ►  public class Abhinav {
825 ►      public static void main(String[] args) {
826          Scanner sc = new Scanner(System.in);
827
828          System.out.print("Enter number of rows: ");
829          int rows = sc.nextInt();
830
831          for (int i = rows; i >= 1; i--) {
832              for (int j = 1; j <= i; j++) {
833                  System.out.print("* ");
834              }
835              System.out.println();
836          }
837
838          sc.close();
839      }
840  }
841
842
843
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 5
* * * *
* * *
* *
* 
*
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

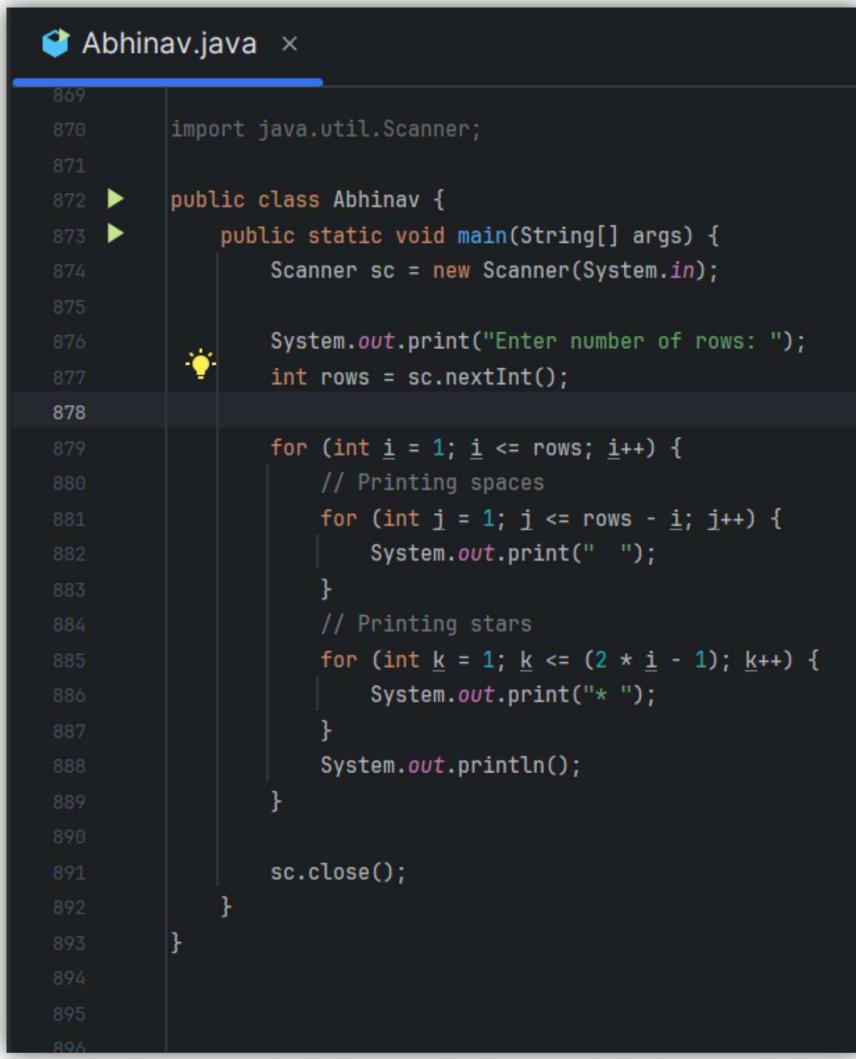
21. Write a program to print Inverted Number Pyramid Pattern.

```
845     import java.util.Scanner;  
846  
847 ►  public class Abhinav {  
848 ►      public static void main(String[] args) {  
849          Scanner sc = new Scanner(System.in);  
850  
851          System.out.print("Enter number of rows: ");  
852          int rows = sc.nextInt();  
853  
854          for (int i = rows; i >= 1; i--) {  
855              for (int j = 1; j <= i; j++) {  
856                  System.out.print(j + " ");  
857              }  
858              System.out.println();  
859          }  
860  
861          sc.close();  
862      }  
863  }  
864  
865  
866
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java  
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav  
Enter number of rows: 5  
1 2 3 4 5  
1 2 3 4  
1 2 3  
1 2  
1  
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

22. Write a program to print star Pyramid Pattern.



```
869
870     import java.util.Scanner;
871
872 ►  public class Abhinav {
873 ►      public static void main(String[] args) {
874          Scanner sc = new Scanner(System.in);
875
876          System.out.print("Enter number of rows: ");
877          int rows = sc.nextInt();
878
879          for (int i = 1; i <= rows; i++) {
880              // Printing spaces
881              for (int j = 1; j <= rows - i; j++) {
882                  System.out.print("  ");
883              }
884              // Printing stars
885              for (int k = 1; k <= (2 * i - 1); k++) {
886                  System.out.print("* ");
887              }
888              System.out.println();
889          }
890
891          sc.close();
892      }
893  }
894
895
896
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 5
*
 *
 *
 *
 *
 *
 *
 *
 *
 *
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

23. Write a program to print Inverted star Pyramid Pattern.

```
Abhinav.java ×

894
895     import java.util.Scanner;
896
897 ►  public class Abhinav {
898 ►      public static void main(String[] args) {
899         Scanner sc = new Scanner(System.in);
900
901         System.out.print("Enter number of rows: ");
902         int rows = sc.nextInt();
903
904         for (int i = rows; i >= 1; i--) {
905             // Printing spaces
906             for (int j = 1; j <= rows - i; j++) {
907                 System.out.print("  ");
908             }
909             // Printing stars
910             for (int k = 1; k <= (2 * i - 1); k++) {
911                 System.out.print("* ");
912             }
913             System.out.println();
914         }
915
916         sc.close();
917     }
918 }
919
920
921
922
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 5
* * * * *
* * * * *
* * * *
* *
*
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

24. Write a program to print Diamond star Pattern.

```
Abhinav.java ×

894
895     import java.util.Scanner;
896
897 ►  public class Abhinav {
898     ►    public static void main(String[] args) {
899         Scanner sc = new Scanner(System.in);
900
901         System.out.print("Enter number of rows: ");
902         int rows = sc.nextInt();
903
904         for (int i = rows; i >= 1; i--) {
905             // Printing spaces
906             for (int j = 1; j <= rows - i; j++) {
907                 System.out.print("  ");
908             }
909             // Printing stars
910             for (int k = 1; k <= (2 * i - 1); k++) {
911                 System.out.print("* ");
912             }
913             System.out.println();
914         }
915
916         sc.close();
917     }
918 }
919
920
921
922
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 5
* * * * * *
* * * * *
* * * *
* *
*
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

25. Write a program to print Half Diamond star Pattern.

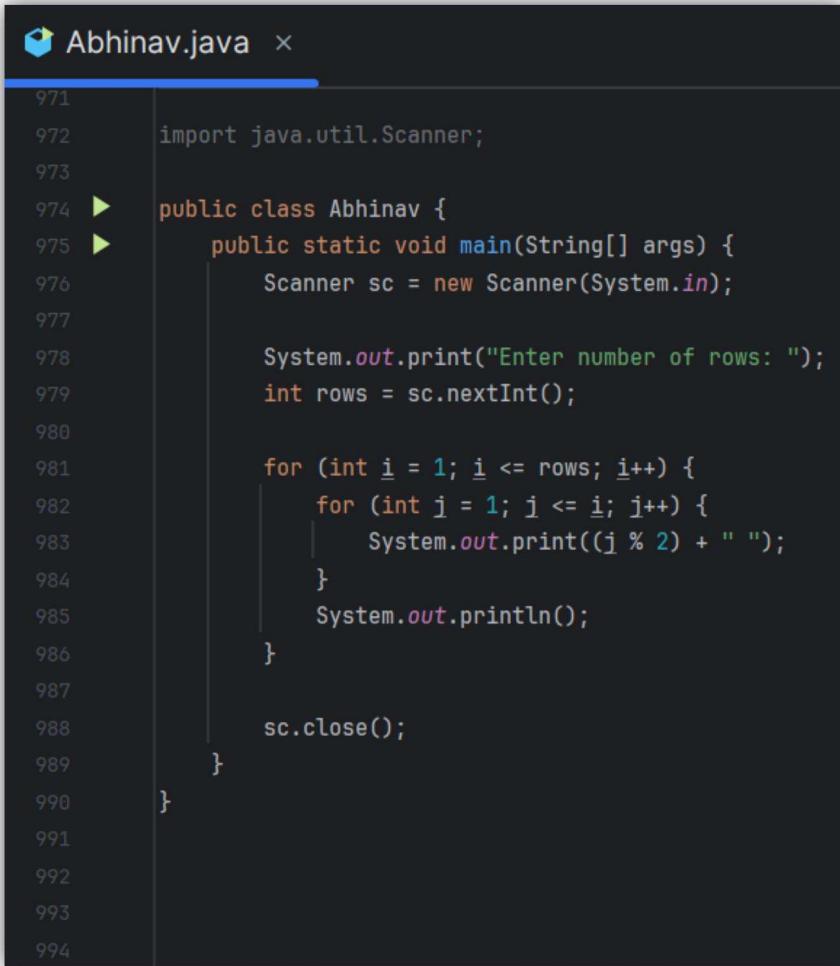
```
Abhinav.java ×

930 import java.util.Scanner;
931
932 ► public class Abhinav {
933 ►     public static void main(String[] args) {
934         Scanner sc = new Scanner(System.in);
935
936         // Taking input for number of rows
937         System.out.print("Enter number of rows: ");
938         int rows = sc.nextInt();
939
940         // Upper half of the diamond
941         for (int i = 1; i <= rows; i++) {
942             for (int j = 1; j <= i; j++) {
943                 System.out.print("* ");
944             }
945             System.out.println();
946         }
947
948         // Lower half of the diamond
949         for (int i = rows - 1; i >= 1; i--) {
950             for (int j = 1; j <= i; j++) {
951                 System.out.print("* ");
952             }
953             System.out.println();
954         }
955
956         sc.close();
957     }
958 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 5
*
* *
* * *
* * * *
* * * * *
* * * *
* *
*
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> []
```

26. Write a program to print Binary Number Triangle Pattern.



```
Abhinav.java  x
971
972     import java.util.Scanner;
973
974 ►  public class Abhinav {
975 ►      public static void main(String[] args) {
976          Scanner sc = new Scanner(System.in);
977
978          System.out.print("Enter number of rows: ");
979          int rows = sc.nextInt();
980
981          for (int i = 1; i <= rows; i++) {
982              for (int j = 1; j <= i; j++) {
983                  System.out.print((j % 2) + " ");
984              }
985              System.out.println();
986          }
987
988          sc.close();
989      }
990  }
991
992
993
994
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 5
1
1 0
1 0 1
1 0 1 0
1 0 1 0 1
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

27. Write a program to print Number crown Pattern.

```
Abhinav.java x
994
995     import java.util.Scanner;
996
997 ► public class Abhinav {
998 ►     public static void main(String[] args) {
999         Scanner sc = new Scanner(System.in);
1000
1001         // Taking input for number of rows
1002         System.out.print("Enter number of rows: ");
1003         int rows = sc.nextInt();
1004
1005         // Printing the number crown pattern
1006         for (int i = 1; i <= rows; i++) {
1007             // Left side numbers
1008             for (int j = 1; j <= i; j++) {
1009                 System.out.print(j + " ");
1010             }
1011
1012             // Spaces in the middle
1013             for (int j = 1; j <= 2 * (rows - i); j++) {
1014                 System.out.print(" ");
1015             }
1016
1017             // Right side numbers
1018             for (int j = i; j >= 1; j--) {
1019                 System.out.print(j + " ");
1020             }
1021
1022             System.out.println();
1023         }
1024
1025         sc.close();
1026     }
1027 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 5
1           1
1 2         2 1
1 2 3       3 2 1
1 2 3 4     4 3 2 1
1 2 3 4 5 5 4 3 2 1
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

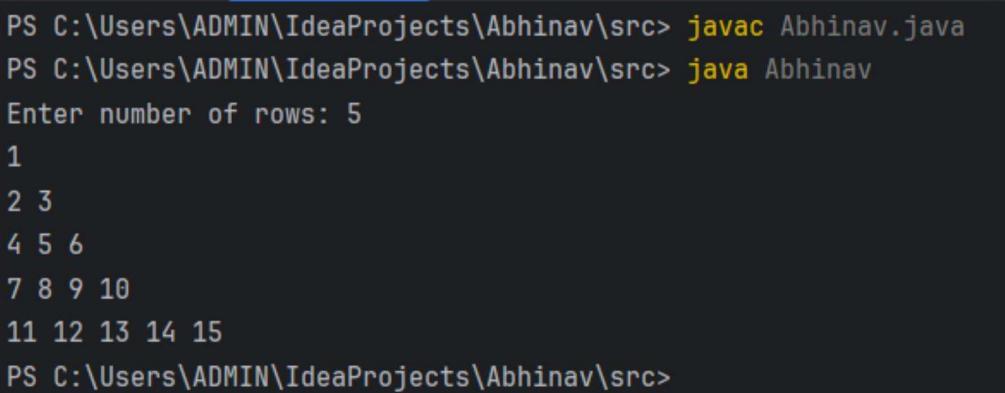
28. Write a program to print Increasing Number Triangle Pattern.



The screenshot shows a code editor window titled "Abhinav.java". The code is a Java program that prints an increasing number triangle pattern. It uses two nested loops: an outer loop for rows and an inner loop for columns. The variable "num" is used to keep track of the current number being printed, starting at 1 and incrementing by 1 for each column in a row. The user is prompted to enter the number of rows. The code is annotated with line numbers from 1039 to 1062. A yellow lightbulb icon is placed next to the line 1050, indicating a potential issue or warning.

```
1039 import java.util.Scanner;
1040
1041 ► public class Abhinav {
1042 ►     public static void main(String[] args) {
1043         Scanner sc = new Scanner(System.in);
1044
1045         // Taking input for number of rows
1046         System.out.print("Enter number of rows: ");
1047         int rows = sc.nextInt();
1048
1049         // Printing the increasing number triangle pattern
1050         int num = 1;
1051         for (int i = 1; i <= rows; i++) {
1052             for (int j = 1; j <= i; j++) {
1053                 System.out.print(num + " ");
1054                 num++;
1055             }
1056             System.out.println();
1057         }
1058
1059         sc.close();
1060     }
1061 }
1062
```

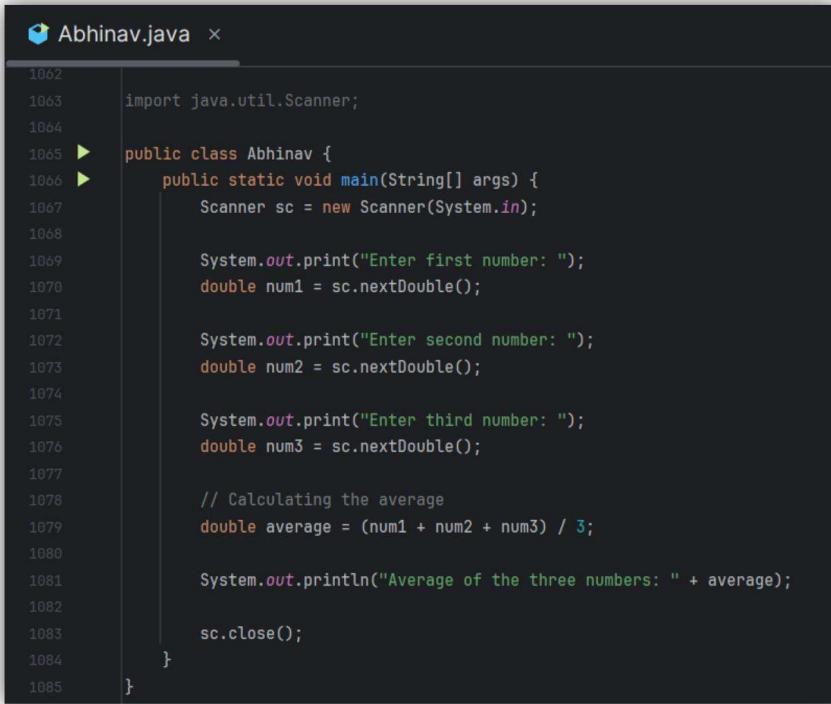
Output –



The screenshot shows a terminal window with the following session:

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter number of rows: 5
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

29. Compute the average of three numbers through a Java Program.



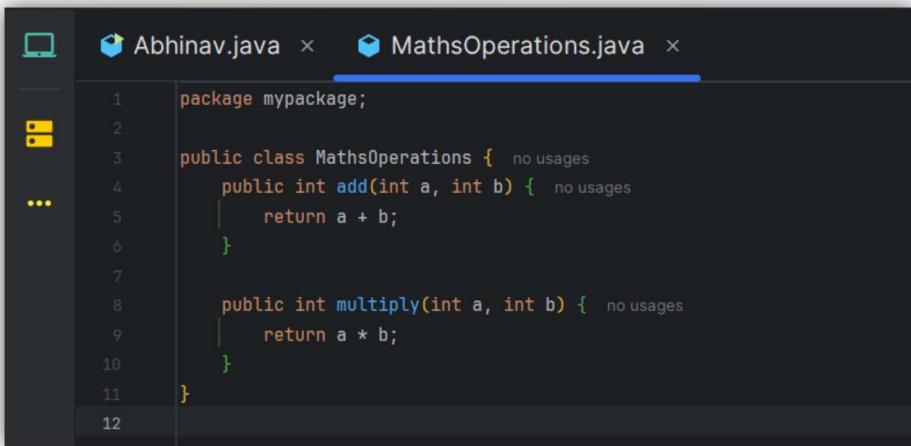
```
1062
1063     import java.util.Scanner;
1064
1065 ► public class Abhinav {
1066 ►     public static void main(String[] args) {
1067         Scanner sc = new Scanner(System.in);
1068
1069         System.out.print("Enter first number: ");
1070         double num1 = sc.nextDouble();
1071
1072         System.out.print("Enter second number: ");
1073         double num2 = sc.nextDouble();
1074
1075         System.out.print("Enter third number: ");
1076         double num3 = sc.nextDouble();
1077
1078         // Calculating the average
1079         double average = (num1 + num2 + num3) / 3;
1080
1081         System.out.println("Average of the three numbers: " + average);
1082
1083         sc.close();
1084     }
1085 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Enter first number: 10
Enter second number: 20
Enter third number: 30
Average of the three numbers: 20.0
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

30. Construct a program to design a package in Java.

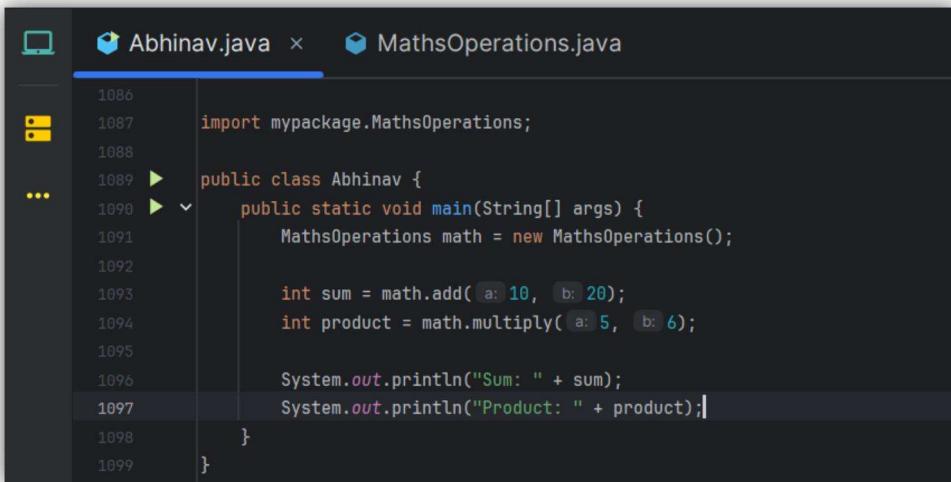
Create a Package (mypackage) ---



A screenshot of an IDE showing the code for `MathsOperations.java`. The code defines a package named `mypackage` and a class `MathsOperations` with two methods: `add` and `multiply`.

```
1 package mypackage;
2
3 public class MathsOperations {
4     public int add(int a, int b) {
5         return a + b;
6     }
7
8     public int multiply(int a, int b) {
9         return a * b;
10    }
11 }
12
```

Use the Package in Main Class ---



A screenshot of an IDE showing the code for `Abhinav.java`. It imports the `MathsOperations` class from the `mypackage` and uses its methods in the `main` method.

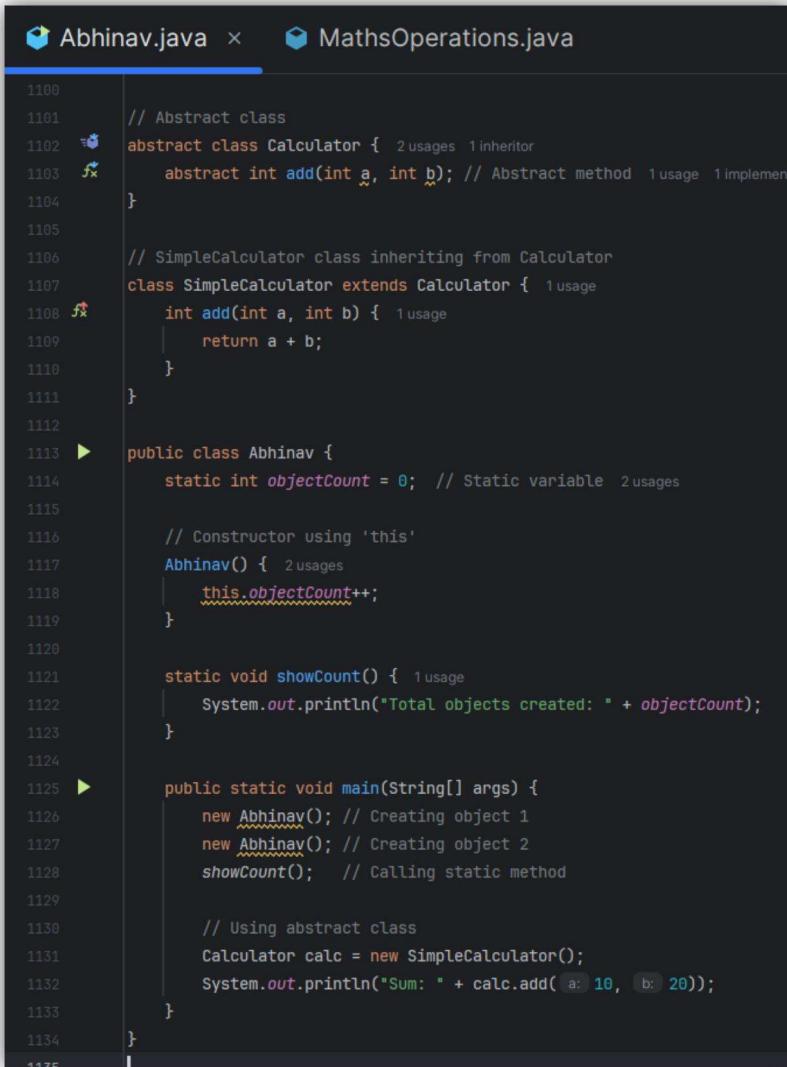
```
1086
1087 import mypackage.MathsOperations;
1088
1089 public class Abhinav {
1090     public static void main(String[] args) {
1091         MathsOperations math = new MathsOperations();
1092
1093         int sum = math.add( 10, 20);
1094         int product = math.multiply( 5, 6);
1095
1096         System.out.println("Sum: " + sum);
1097         System.out.println("Product: " + product);
1098     }
1099 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac -d . mypackage/MathsOperations.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Sum: 30
Product: 30
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> []
```

31. Write a Java program that demonstrates the use of keywords:

(a) this (b) static (c) abstract

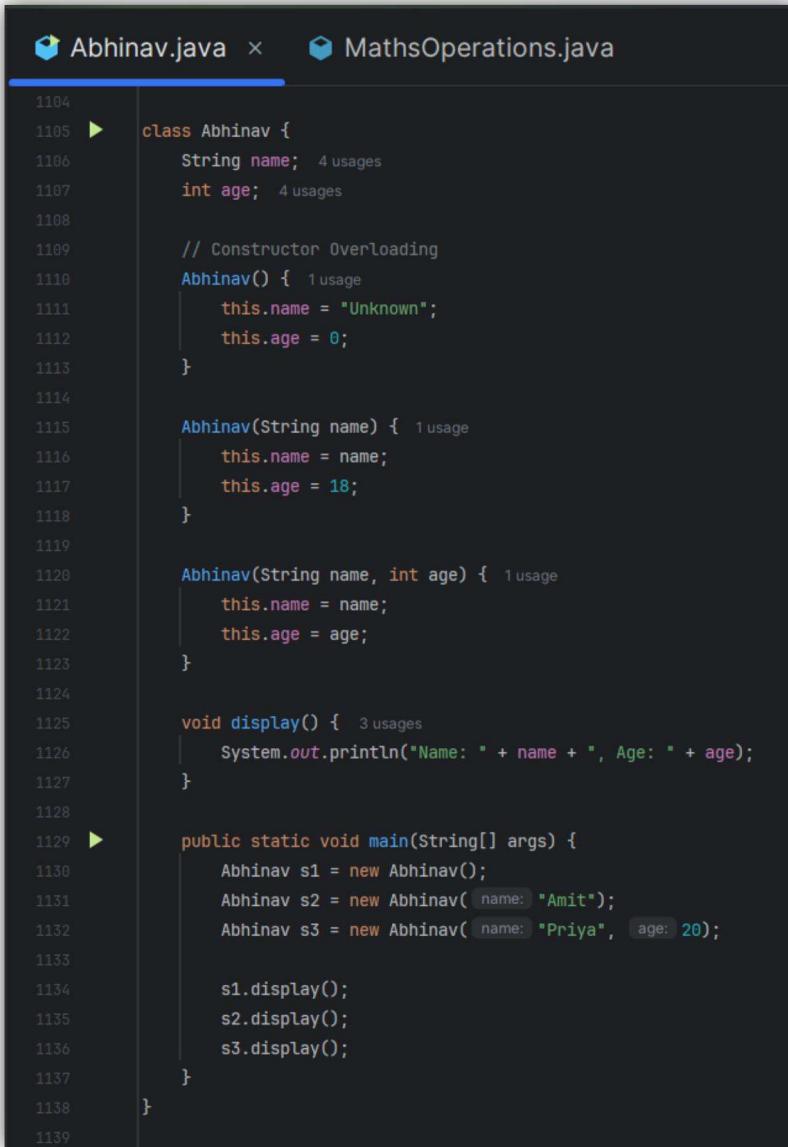


```
1100
1101 // Abstract class
1102 abstract class Calculator { 2 usages 1 inheritor
1103     abstract int add(int a, int b); // Abstract method 1 usage 1 implementor
1104 }
1105
1106 // SimpleCalculator class inheriting from Calculator
1107 class SimpleCalculator extends Calculator { 1 usage
1108     int add(int a, int b) { 1 usage
1109         return a + b;
1110     }
1111 }
1112
1113 public class Abhinav {
1114     static int objectCount = 0; // Static variable 2 usages
1115
1116     // Constructor using 'this'
1117     Abhinav() { 2 usages
1118         this.objectCount++;
1119     }
1120
1121     static void showCount() { 1 usage
1122         System.out.println("Total objects created: " + objectCount);
1123     }
1124
1125     public static void main(String[] args) {
1126         new Abhinav(); // Creating object 1
1127         new Abhinav(); // Creating object 2
1128         showCount(); // Calling static method
1129
1130         // Using abstract class
1131         Calculator calc = new SimpleCalculator();
1132         System.out.println("Sum: " + calc.add( a: 10, b: 20));
1133     }
1134 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Total objects created: 2
Sum: 30
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```

32. WAP that demonstrates the use of constructor overloading concepts.

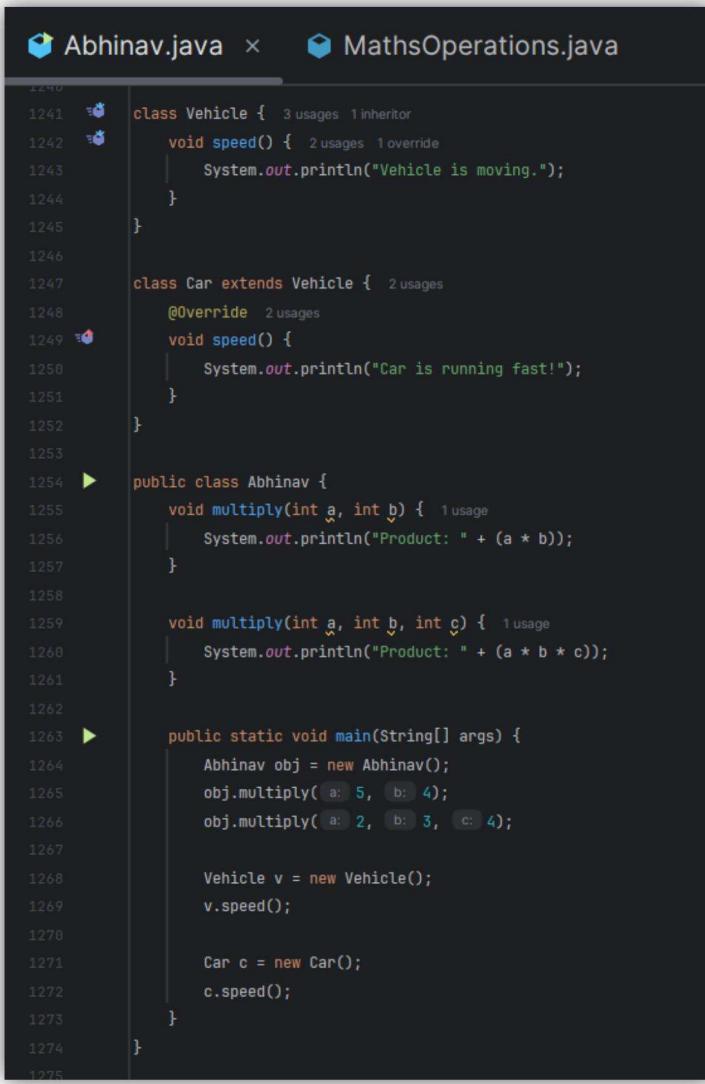


```
1104
1105 ►  class Abhinav {
1106     String name;  4 usages
1107     int age;  4 usages
1108
1109     // Constructor Overloading
1110     Abhinav() { 1 usage
1111         this.name = "Unknown";
1112         this.age = 0;
1113     }
1114
1115     Abhinav(String name) { 1 usage
1116         this.name = name;
1117         this.age = 18;
1118     }
1119
1120     Abhinav(String name, int age) { 1 usage
1121         this.name = name;
1122         this.age = age;
1123     }
1124
1125     void display() { 3 usages
1126         System.out.println("Name: " + name + ", Age: " + age);
1127     }
1128
1129 ►  public static void main(String[] args) {
1130     Abhinav s1 = new Abhinav();
1131     Abhinav s2 = new Abhinav( name: "Amit");
1132     Abhinav s3 = new Abhinav( name: "Priya",  age: 20);
1133
1134     s1.display();
1135     s2.display();
1136     s3.display();
1137 }
1138 }
1139 }
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Name: Unknown, Age: 0
Name: Amit, Age: 18
Name: Priya, Age: 20
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> []
```

33. WAP that exhibits the method overriding and method overloading.



The screenshot shows an IDE interface with two tabs: "Abhinav.java" and "MathsOperations.java". The "Abhinav.java" tab is active, displaying the following code:

```
1240
1241     class Vehicle { 3 usages 1 inheritor
1242         void speed() { 2 usages 1 override
1243             System.out.println("Vehicle is moving.");
1244         }
1245     }
1246
1247     class Car extends Vehicle { 2 usages
1248         @Override 2 usages
1249         void speed() {
1250             System.out.println("Car is running fast!");
1251         }
1252     }
1253
1254     public class Abhinav {
1255         void multiply(int a, int b) { 1 usage
1256             System.out.println("Product: " + (a * b));
1257         }
1258
1259         void multiply(int a, int b, int c) { 1 usage
1260             System.out.println("Product: " + (a * b * c));
1261         }
1262
1263     public static void main(String[] args) {
1264         Abhinav obj = new Abhinav();
1265         obj.multiply( 5, 4);
1266         obj.multiply( 2, 3, 4);
1267
1268         Vehicle v = new Vehicle();
1269         v.speed();
1270
1271         Car c = new Car();
1272         c.speed();
1273     }
1274 }
1275
```

Output –

```
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> javac Abhinav.java
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src> java Abhinav
Product: 20
Product: 24
Vehicle is moving.
Car is running fast!
PS C:\Users\ADMIN\IdeaProjects\Abhinav\src>
```