Programming in Java

BCA-DS-402

Manav Rachna International Institute of Research and Studies School of Computer Applications

Department of Computer Applications

Submitted By	
Student Name	Harshit Sehrawat
Roll No	23/SCA/BCA(AI&ML)/019
Programme	Bachelor of Computer Applications
Semester	4th Semester
Section/Group	IV -C
Department	School of Computer Applications
Batch	2023-26
Submitted To	
Faculty Name	Dr. Priyanka Sharma

MANAV RACHNA
vidyanatariksha
Manav Rachna International Institute of Research and Studies
Deemed to be University under section 3 of the UGC Act, 1956

SCHOOL OF COMPUTER APPLICATIONS

S.No	Date	**Aim of the programme**	**Signature/ Date**
session-1			
1		Write java program to print hello world	
2		Java Program to take input from user and print the sum of two numbers.	
3		Create a java program to check whether a number entered by user is even or odd	
4		Create a java program to print the average and sum of 5 numbers entered by user.	
5		Program to calculate the factorial of a number	
6		Program to print Fibonacci series up to n terms	
session-2			
1		Program to reverse a number	
2		Program to check if a number is a palindrome	
3		Program for a simple calculator	
4		Program to check if a number is prime	
5		Program to check if a number is an Armstrong number	
session-3			
1		Print Multiplication Table	
2		Calculate Sum and Average of Array Elements	
3		Reverse a String	
4		Find Factorial of a Number Using Recursion	
5		Sort an Array in Ascending Order	

session-4			
1		Check Palindrome for a String	
2		Count Vowels and Consonants in a String	
3		Write a program to demonstrate type casting.	
4		Write a program to generate prime numbers between 1 & given number	
session-5			
1		Program to Demonstrate a Simple Class with Methods	
2		Program for Class with Parameterized Constructor	
3		Program to Find the Area of a Rectangle Using Methods	
4		Program for Bank Account Class with Deposit and Withdraw Methods	
5		Program to Demonstrate Method Overloading	
session-6			
1		Program to Demonstrate Static Methods	
2		Program to Demonstrate Method Overriding	
3		Program to Demonstrate Getters and Setters	
4		Program to Demonstrate a Class with Multiple Methods	
5		Program to Demonstrate Object Passing in Methods	
6		Write a program to create a simple class to find out the area and perimeter of rectangle using super and this keyword.	
7		Write a program to count the number of objects created for a class using static member function	

LAB SESSION 1

Q1 Write java program to print hello world

Main.java

Share Run

```java
// Online Java Compiler
// Use this editor to write, compile and run your Java code online

class Main {
    public static void main(String[] args) {
        System.out.println("hellow world");
    }
}
```

Output

Output Clear

```
hellow world

=== Code Execution Successful ===
```

O2 - Java Program to take input from user and print the sum of two numbers

```java
// Online Java Compiler
// Use this editor to write, compile and run your Java code online
import java.util.Scanner;
class Main {
    public static void main(String[] args) {

    int x, y, sum;
    Scanner myObj = new Scanner(System.in);
    System.out.println("Type a number:");
    x = myObj.nextInt();

    System.out.println("Type another number:");
    y = myObj.nextInt();

    sum = x + y;
    System.out.println("Sum is: " + sum);
    }
}
```

Output-

Output

Clear

```
Type a number:
22
Type another number:
10
Sum is: 32

=== Code Execution Successful ===
```

Q3 - Create a java program to check whether a number entered by user is even or odd

Main.java

Share Run

```java
// Online Java Compiler
// Use this editor to write, compile and run your Java code online
import java.util.Scanner;
class Main {
    public static void main(String[] args) {

        Scanner reader = new Scanner(System.in);

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

        if(num % 2 == 0)
            System.out.println(num + " is even");
        else
            System.out.println(num + " is odd");
    }
}
```

Output –

Output

Clear

```
Enter a number: 22
22 is even

=== Code Execution Successful ===
```

Q4 -Create a java program to print the average and sum of 5 numbers entered by user.

Main.java

Share Run

```
3  import java.util.Scanner;
4  class Main {
5      public static void main(String[] args) {
6
7          int i,n=0,s=0;
8      double avg;
9      {
10
11         System.out.println("Input the 5 numbers : ");
12
13     }
14         for (i=0;i<5;i++)
15         {
16             Scanner in = new Scanner(System.in);
17             n = in.nextInt();
18
19         s +=n;
20     }
21     avg=s/5;
22     System.out.println("The sum of 5 no is : " +s+"\nThe
```

Output

Output Clear

```
Input the 5 numbers :
22
10
29
20
19
The sum of 5 no is : 90
The Average is : 18.0

=== Code Execution Successful ===
```

Q5 - Program to calculate the factorial of a number

Main.java

Share Run

```java
// Online Java Compiler
// Use this editor to write, compile and run your Java code online
import java.util.Scanner;
class Main {
    public static void main(String[] args) {


        int num = 10;
        long factorial = 1;
        for(int i = 1; i <= num; ++i)
        {
            // factorial = factorial * i;
            factorial *= i;
        }
        System.out.printf("Factorial of %d = %d", num, factorial
            );
    }
}
```

Output –

Output Clear

```
Factorial of 10 = 3628800
=== Code Execution Successful ===
```

Q6 - Program to print Fibonacci series up to n terms

```
import java.io.*;
class GFG {
    static void Fibonacci(int N)
    {
        int num1 = 0, num2 = 1;
        for (int i = 0; i < N; i++) {
            System.out.print(num1 + " ")
            int num3 = num2 + num1;
            num1 = num2;
            num2 = num3;
        }
    }
    public static void main(String args[])
    {
        int N = 10;
        Fibonacci(N);
    }
}
```

Output

```
Output:

0 1 1 2 3 5 8 13 21 34
```

----end------

LAB SESSION – 2

Q1- Program to reverse a number

Main.java

Share Run

```java
class Main {
  public static void main(String[] args) {
    int num = 1234, reversed = 0;
    System.out.println("Original Number: " + num);
    while(num != 0) {
      int digit = num % 10;
      reversed = reversed * 10 + digit;
      num /= 10;
    }
    System.out.println("Reversed Number: " + reversed);
  }
}
```

Output-

Output Clear

```
Original Number: 1234
Reversed Number: 4321

=== Code Execution Successful ===
```

Q2 -Program to check if a number is a palindrome

Main.java

Share Run

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

    String str = "Radar", reverseStr = "";

    int strLength = str.length();

    for (int i = (strLength - 1); i >=0; --i) {
      reverseStr = reverseStr + str.charAt(i);
    }

    if (str.toLowerCase().equals(reverseStr.toLowerCase())) {
      System.out.println(str + " is a Palindrome String.");
    }
    else {
      System.out.println(str + " is not a Palindrome String.");
    }
  }
}
```

Output –

Output Clear

```
Radar is a Palindrome String.

=== Code Execution Successful ===
```

Q3 – Program for a simple calculator

Main.java

Share Run

```java
import java.util.Scanner;

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

    char operator;
    Double number1, number2, result;
    Scanner input = new Scanner(System.in);
    System.out.println("Choose an operator: +, -, *, or /");
    operator = input.next().charAt(0);
    System.out.println("Enter first number");
    number1 = input.nextDouble();
    System.out.println("Enter second number");
    number2 = input.nextDouble();
    switch (operator) {
      case '+':
        result = number1 + number2;
        System.out.println(number1 + " + " + number2 + " = " +
            result);
        break;
```

Output –

Output Clear

```
Choose an operator: +, -, *, or /
+
Enter first number
23
Enter second number
34
23.0 + 34.0 = 57.0

=== Code Execution Successful ===
```

Q4 - Program to check if a number is prime

Main.java

Share Run

```java
public class Main {
  public static void main(String[] args) {
    int num = 29;
    boolean flag = false;
    if (num == 0 || num == 1) {
        flag = true;
    }
    for (int i = 2; i <= num / 2; ++i) {
      if (num % i == 0) {
        flag = true;
        break;
      }
    }
    if (!flag)
      System.out.println(num + " is a prime number.");
    else
      System.out.println(num + " is not a prime number.");
  }
}
```

Output –

Output Clear

```
29 is a prime number.

=== Code Execution Successful ===
```

Q5-Program to check if a number is an Armstrong number

Main.java

Share Run

```
5   public static void main(String[] args) {
6       int number = 371, originalNumber, remainder, result = 0
            ;
7
8       originalNumber = number;
9
10      while (originalNumber != 0)
11      {
12          remainder = originalNumber % 10;
13          result += Math.pow(remainder, 3);
14          originalNumber /= 10;
15      }
16
17      if(result == number)
18          System.out.println(number + " is an Armstrong
                number.");
19      else
20          System.out.println(number + " is not an Armstrong
                number.");
```

Output –

Output Clear

```
371 is an Armstrong number.

=== Code Execution Successful ===
```

LAB SESSION – 3

1- Print Multiplication Table

Main.java

Share Run

```java
import java.util.Scanner;

public class Main {
    public static void multiplicationTable(int n) {
        for (int i = 1; i <= 10; i++) {
            System.out.println(n + " x " + i + " = " + (n * i));
        }
    }

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

Output

```
Enter a number: 45
45 x 1 = 45
45 x 2 = 90
45 x 3 = 135
45 x 4 = 180
45 x 5 = 225
45 x 6 = 270
45 x 7 = 315
45 x 8 = 360
45 x 9 = 405
45 x 10 = 450

=== Code Execution Succes
```

2- Calculate Sum and Average of Array Elements

Main.java

Share Run

```java
import java.util.Scanner;

public class Main {
    public static void sumAndAverage(int[] arr) {
        int sum = 0;
        for (int num : arr) {
            sum += num;
        }
        double average = (arr.length > 0) ? (double) sum / arr.length
            : 0;
        System.out.println("Sum: " + sum + ", Average: " + average);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter numbers separated by space: ");
        String[] input = sc.nextLine().split(" ");
        int[] arr = new int[input.length];
        for (int i = 0; i < input.length; i++) {
            arr[i] = Integer.parseInt(input[i]);
        }
        sumAndAverage(arr);
    }
}
```

Output

```
Enter numbers separated by space: 3 5 6
Sum: 14, Average: 4.666666666666667

=== Code Execution Successful ===
```

3- Reverse a String

Main.java

Share Run

Output

```java
import java.util.Scanner;

public class Main {
    public static String reverseString(String s) {
        StringBuilder reversed = new StringBuilder(s);
        return reversed.reverse().toString();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String s = sc.nextLine();
        System.out.println(reverseString(s));
    }
}
```

```
Enter a string: hieharshit
tihsraheih

=== Code Execution Successful
```

4- Find Factorial of a Number Using Recursion

Main.java

Share Run

Output

```java
import java.util.Scanner;

public class Main {
    public static int factorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        }
        return n * factorial(n - 1);
    }

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

```
Enter a number: 5
120

=== Code Execution
```

5- Sort an Array in Ascending Order

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

public class Main {
    public static void sortArray(int[] arr) {
        Arrays.sort(arr);
        System.out.println(Arrays.toString(arr));
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter numbers separated by space: ");
        String[] input = sc.nextLine().split(" ");
        int[] arr = new int[input.length];
        for (int i = 0; i < input.length; i++) {
            arr[i] = Integer.parseInt(input[i]);
        }
        sortArray(arr);
    }
}

```

```
Enter numbers separated by space: 3 5 6 7
[3, 5, 6, 7]

=== Code Execution Successful ===
```

LAB SESSION – 4

1- Check Palindrome for a String

Main.java

```java
import java.util.Scanner;

public class Main {
    public static boolean isPalindrome(String s) {
        String reversed = new StringBuilder(s).reverse().toString();
        return s.equals(reversed);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String s = sc.nextLine();
        if (isPalindrome(s)) {
            System.out.println(s + " is a palindrome.");
        } else {
            System.out.println(s + " is not a palindrome.");
        }
    }
}
```

```
Enter a string: ma'am
ma'am is a palindrome.

=== Code Execution Succe
```

2- Count Vowels and Consonants in a String

```java
import java.util.Scanner;
public class Main {
    public static void countVowelsAndConsonants(String s) {
        int vowels = 0, consonants = 0;
        s = s.toLowerCase();
        for (char c : s.toCharArray()) {
            if (Character.isLetter(c)) {
                if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c
                    == 'u') {
                    vowels++;
                } else {
                    consonants++;
                }
            }
        }
        System.out.println("Vowels: " + vowels + ", Consonants: " +
            consonants);
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String s = sc.nextLine();
        countVowelsAndConsonants(s);
    }
}
```

```
Enter a string: hie harshit
Vowels: 4, Consonants: 6

=== Code Execution Successful
```

3- Write a program to demonstrate type casting.

```java
public class Main {
    public static void main(String[] args) {
        int intVal = 10;
        double doubleVal = intVal; // Implicit type casting (Widening)
        System.out.println("Implicit type casting (int to double): " +
            doubleVal);

        double anotherDoubleVal = 10.5;
        int anotherIntVal = (int) anotherDoubleVal; // Explicit type
            casting (Narrowing)
        System.out.println("Explicit type casting (double to int): " +
            anotherIntVal);
    }
}
```

```
Implicit type casting (int to double): 10.0
Explicit type casting (double to int): 10

=== Code Execution Successful ===
```

4- Write a program to generate prime numbers between 1 & given number

```java
import java.util.Scanner;
public class Main {
    public static boolean isPrime(int num) {
        if (num <= 1) {
            return false; }
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false; } }
        return true; }
    public static void generatePrimes(int n) {
        for (int i = 2; i <= n; i++) {
            if (isPrime(i)) {
                System.out.println(i);}
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n = sc.nextInt();
        System.out.println("Prime numbers between 1 and " + n + " are
            :");
        generatePrimes(n);
    }
}
```

```
Enter a number: 69
Prime numbers between 1 and 69 are:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67

=== Code Execution Successful ===
```

LAB SESSION - 5

1. Program to Demonstrate a Simple Class with Method

Main.java | Share | Run | Output

```
class Main {
    void display() {
        System.out.println("This is a simple class method.");
    }

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

```
This is a simple class method.

=== Code Execution Successful ===
```

2. Program for Class with Parameterized Constructor

Main.java | Share | Run | Output

```
class Main {
    String name;
    int age;
    Main(String n, int a) {
        name = n;
        age = a;
    }
    void display() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
    public static void main(String[] args) {
        Main s1 = new Main("Harshit", 21);
        s1.display();
    }
}
```

```
Name: Harshit, Age: 21

=== Code Execution Successful ===
```

3. Program to Find the Area of a Rectangle Using Methods

Main.java

Share Run

Output

```
class Main {
    double length, width;

    void setValues(double l, double w) {
        length = l;
        width = w;
    }
    double area() {
        return length * width;
    }
    public static void main(String[] args) {
        Main rect = new Main();
        rect.setValues(10.5, 5.2);
        System.out.println("Area of Rectangle: " + rect.area
            ());
    }
}
```

```
Area of Rectangle: 54.6

=== Code Execution Successful ===
```

4. Program for Bank Account Class with Deposit and Withdraw Methods

Online Java Compiler

Courses by Programiz

Main.java

Share Run

Output

```
class Main {
    double balance;
    Main(double initialBalance) {
        balance = initialBalance;
    }
    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount + ", New Balance:
            " + balance);
    }
    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount + ", New
                Balance: " + balance);
        } else {
            System.out.println("Insufficient funds!");
        }
    }
    public static void main(String[] args) {
        Main acc = new Main(5000);
        acc.deposit(1500);
        acc.withdraw(2000);
        acc.withdraw(5000); // Insufficient funds case
    }
}
```

```
Deposited: 1500.0, New Balance: 6500.0
Withdrawn: 2000.0, New Balance: 4500.0
Insufficient funds!

=== Code Execution Successful ===
```

5. Program to Demonstrate Method Overloading

Main.java

Share Run

Output

```java
class Main {
    int add(int a, int b) {
        return a + b;
    }
    double add(double a, double b) {
        return a + b;
    }
    int add(int a, int b, int c) {
        return a + b + c;
    }
    public static void main(String[] args) {
        Main obj = new Main();
        System.out.println("Sum (int): " + obj.add(5, 10));
        System.out.println("Sum (double): " + obj.add(5.5, 10.3));
        System.out.println("Sum (three ints): " + obj.add(5, 10, 15));
    }
}

```

```
Sum (int): 15
Sum (double): 15.8
Sum (three ints): 30

=== Code Execution Successful ===
```

SESSION 6

1. Program to Demonstrate static method.

class StaticExample.java 3 ✕ Extension: Extension Pack for Java

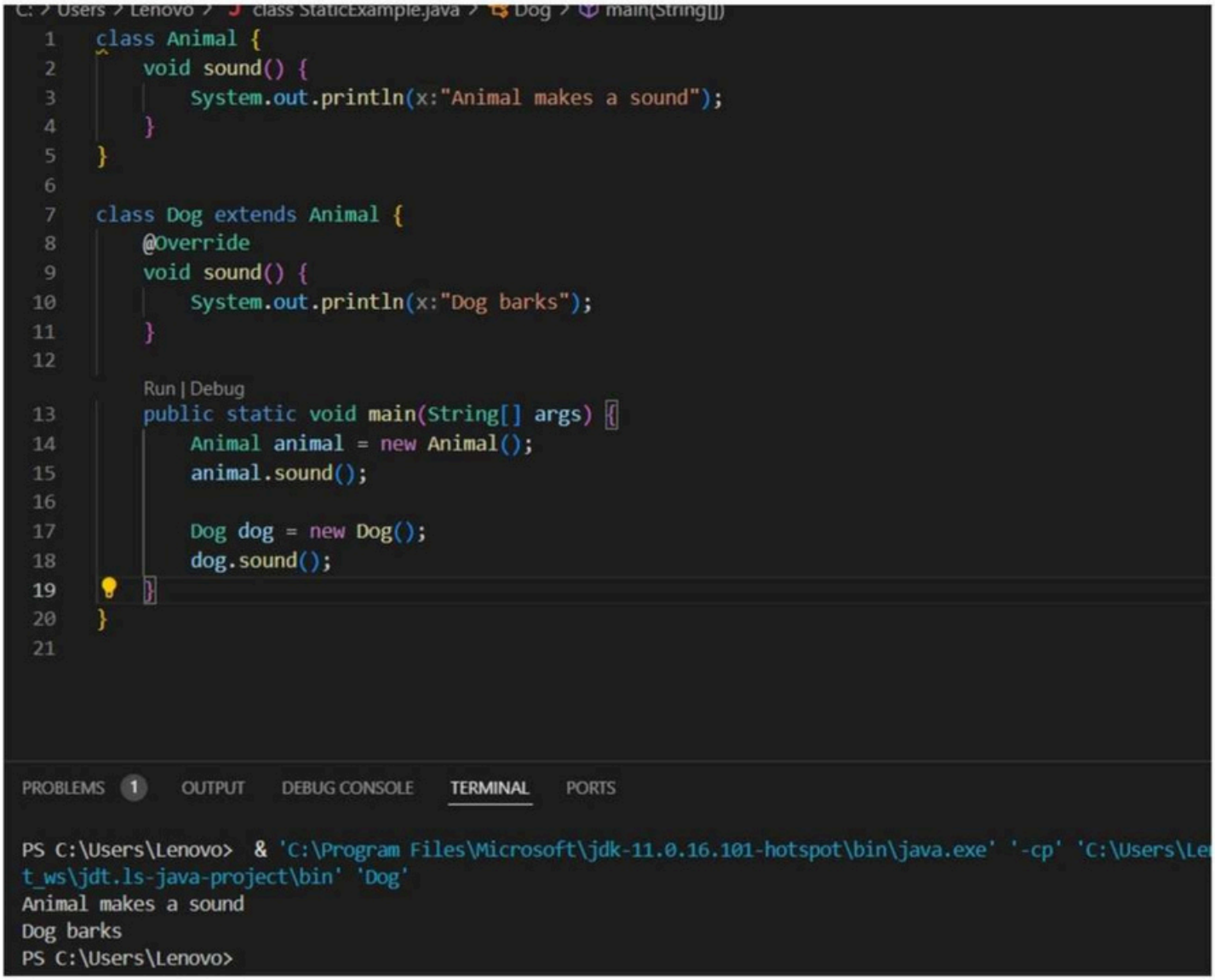
C: > Users > Lenovo > class StaticExample.java > ...

```java
class StaticExample {
    static int count = 0;

    StaticExample() {
        count++;
    }

    static void displayCount() {
        System.out.println("Number of objects created: " + count);
    }

    Run | Debug
    public static void main(String[] args) {
        StaticExample obj1 = new StaticExample();
        StaticExample obj2 = new StaticExample();
        StaticExample.displayCount();
    }
}

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Lenovo> & 'C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\java.exe' '-cp' 'C:\Users\Lenovo\AppDat
t_ws\jdt.ls-java-project\bin' 'StaticExample'
Number of objects created: 2
PS C:\Users\Lenovo>
```

Program to Demonstrate Method Overridi



3. Program to demonstrate getters and setters s

C: > Users > Lenovo > class StaticExample.java > Person > setAge(int)

```java
class Person {
    private String name;
    private int age;
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public static void main(String[] args) {
        Person person = new Person();
        person.setName(name:"Harshit");
        person.setAge(age:20);
        System.out.println("Name: " + person.getName());
        System.out.println("Age: " + person.getAge());
    }
}
```

Run | Debug

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

hit

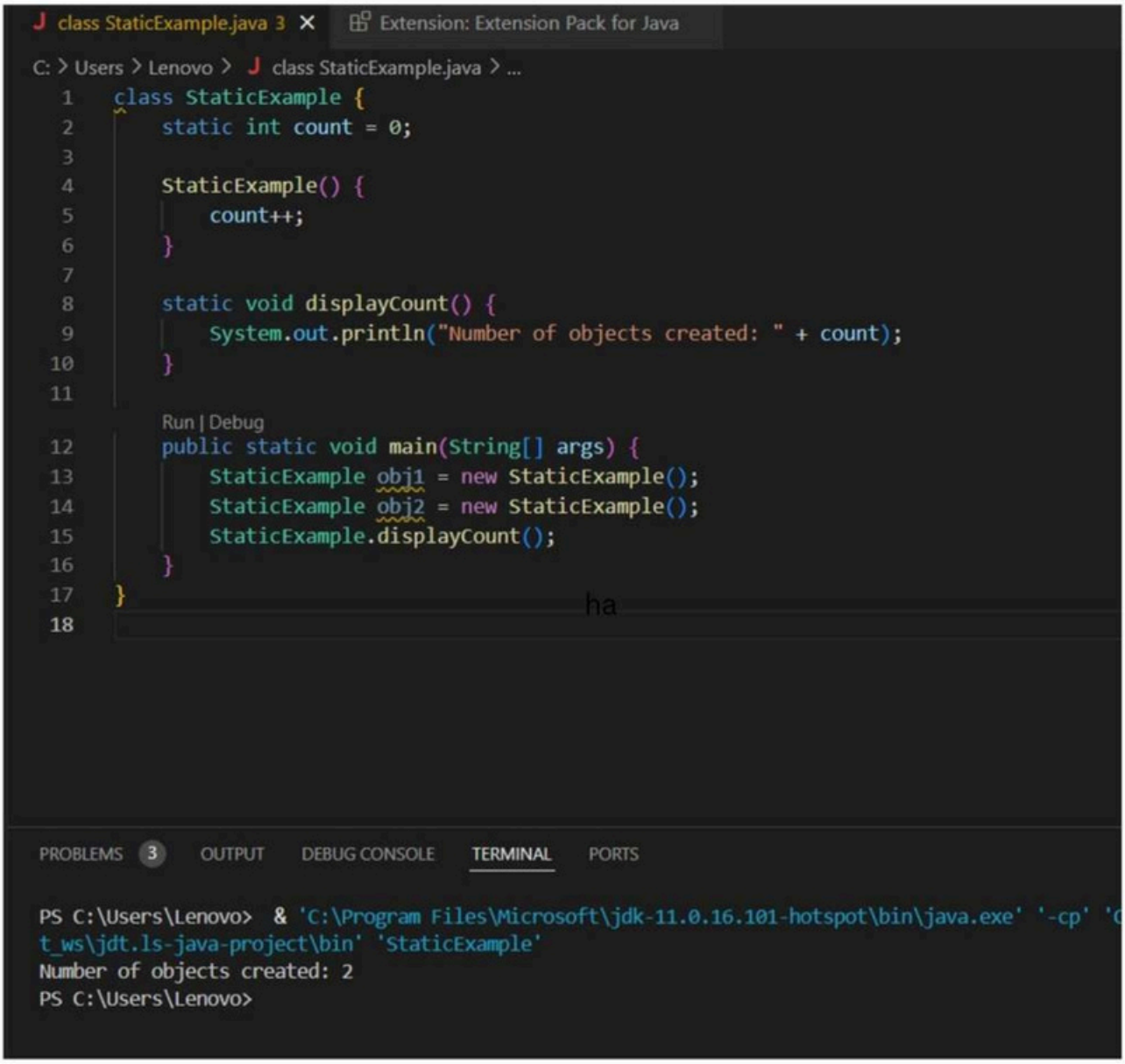
```
t_ws\jdt.ls-java-project\bin' 'Person'
Name: John
Age: 30
PS C:\Users\Lenovo> ^C
PS C:\Users\Lenovo>
PS C:\Users\Lenovo>  & 'C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\java.exe' '-cp' 'C:\Users\Lenovo\Ap
project\bin' 'Person'
Name: Harshit
Age: 20
PS C:\Users\Lenovo>
```

4. Page to demonstrate a class with multiple methods

```java
class Animal {
    void sound() {
        System.out.println(x:"Animal makes a sound");
    }
}

class Dog extends Animal {
    @Override
    void sound() {
        System.out.println(x:"Dog barks");
    }

    Run | Debug
    public static void main(String[] args) {
        Animal animal = new Animal();
        animal.sound();

        Dog dog = new Dog();
        dog.sound();
    }
}
```

```
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Lenovo>  & 'C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\java.exe' '-
t_ws\jdt.ls-java-project\bin' 'Dog'
Animal makes a sound
Dog barks
PS C:\Users\Lenovo>
```

5. Program to Demonstrate Method Object passing in m weo



Abstract Class Example:

```
abstract class Shape {
    abstract void draw();
}

class Circle extends Shape {
    void draw() {
        System.out.println("Drawing Circle");
    }
}

public class AbstractDemo {
    public static void main(String[] args) {
        Shape s = new Circle();
        s.draw();
    }
}                   / /
O u t p u t :
D r a w i n g
C i r c l e
```

Multilevel Inheritance:

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

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

class Puppy extends Dog {
    void weep() {
        System.out.println("Puppy weeps");
    }
}

public class Multilevel {
    public static void main(String[] args) {
        Puppy p = new Puppy();
        p.eat();      p.bark();
        p.weep();

    }
}      / /      O u t p u t :
T h i s          a n i m a l
e a t s    f o o d
//         Dog barks
//         Puppy weeps
```

Multiple Inheritance using Interface:

```
interface Printable {
    void print();
}

interface Showable {
    void show();
}

class A implements Printable, Showable {
    public void print() {
        System.out.println("Print method");
    }

    public void show() {
        System.out.println("Show method");
    }
}

public class MultipleInheritance {
    public static void main(String[] args) {
        A obj = new A();
        obj.print();
        obj.show();
    }
}                  / /
O u t p u t :
P / r i n t
m e t h o d
```

Show method

Partial Interface Implementation:

```
interface Vehicle {
    void start();
    void stop();
}

abstract class Car implements Vehicle {
    public void start() {
        System.out.println("Car starts");
    }
}

class Honda extends Car {
    public void stop() {
        System.out.println("Car stops");
    }
}

public class PartialInterface {
    public static void main(String[] args) {
        Honda h = new Honda();
```

```
        h.start();
        h.stop();
    }
}               / /
O u t p u t :
C a r
/ / s t a r t s     Car stops
```

String Operations Class:

```
class MyString {
    String str;

    MyString(String str) {
        this.str = str;
    }

    boolean isEqual(String other) {
        return str.equals(other);
    }

    String reverse() {
        return new StringBuilder(str).reverse().toString();
    }

    String changeCase() {
        StringBuilder sb = new StringBuilder();
        for (char ch : str.toCharArray()) {
            if (Character.isUpperCase(ch))
                sb.append(Character.toLowerCase(ch));
            else
                sb.append(Character.toUpperCase(ch));
        }
        r e t u r n   s b . t o S t r i n g ( ) ;
    }
}


public class StringDemo {
    public static void main(String[] args) {
        MyString s = new MyString("Java123");
        System.out.println(s.isEqual("Java123")); // true
        System.out.println(s.reverse()); // 321avaJ
        System.out.println(s.changeCase()); // jAVA123
    }
}
```

Exception Handling with Multiple Catch:

```
public class MultiCatch {
    public static void main(String[] args) {
        try {
            int a = 10 / 0;
            String s = null;
```

```
            System.out.println(s.length());
        } catch (ArithmeticException e) {
            System.out.println("Arithmetic Exception occurred");
        } catch (NullPointerException e) {
            System.out.println("Null Pointer Exception occurred");
        }
    }
}         / /         O u t p u t :
A r i t h m e t i c
E x c e p t i o n
o c c u r r e d
```

Package Example - Accessing Members:

```
// File: mypack/ExternalClass.java
package mypack;

public class ExternalClass {
    public void display() {
        System.out.println("External class method");
    }
}

// File: mypack/SamePackage.java
package mypack;

public class SamePackage {
    public static void main(String[] args) {
        ExternalClass ec = new ExternalClass();
        ec.display();
    }
}      / /      O u t p u t :
E x t e r n a l
c l a s s   m e t h o d
```

Import User Defined Package:

```
// File: mypack/Access.java
package mypack;

public class Access {
    public int x = 10;
}

// File: TestPackage.java
import mypack.Access;

public class TestPackage {
    public static void main(String[] args) {
        Access a = new Access();
        System.out.println("Value: " + a.x);
    }
}              / /
O u t p u t :
V a l u e :
1 0
```

User Defined Exception with throw:

```
class MyException extends Exception {
    MyException(String msg) {
        super(msg);
    }
}

public class ThrowExample {
    static void checkAge(int age) throws MyException {
        if (age < 18)
            throw new MyException("Age is below 18");
        else
            System.out.println("Eligible");
    }

    public static void main(String[] args) {
        try {
            checkAge(16);
        } catch (MyException e) {
            System.out.println(e.getMessage());
        }
    }
}                    / /
O u t p u t :    A g e
i s   b e l o w   1 8
```

String Methods (substring, replace, split):

```
public class StringMethods {
    public static void main(String[] args) {
    String str = "Java Programming";
    System.out.println(str.substring(5)); // Programming
    System.out.println(str.replace("Java", "Python")); // Python Programming
    String[] words = str.split(" ");
    for (String word : words) {
            System.out.println(word);
        }
    }
}    // Output: Programming, Python Programming, Java, Programming
```

Custom Exception AgeException:

```
class AgeException extends Exception {
    AgeException(String msg) {
        super(msg);
    }
}

public class AgeCheck {
    static void validateAge(int age) throws AgeException {
        if (age < 18)
            throw new AgeException("Age must be 18 or above");
        else
            System.out.println("Age is valid");
    }

    public static void main(String[] args) {
        try {
            validateAge(17);
        } catch (AgeException e) {
            System.out.println(e.getMessage());
        }
    }
}      // Output:
Age must be 18
or above
```

String Handling and Functions:

```
public class StringFunctions {
    public static void main(String[] args) {
    String str = " Hello Java World ";
    System.out.println("Length: " + str.length());
    System.out.println("Concat: " + str.concat("!!!"));
    System.out.println("Char at 6: " + str.charAt(6));
    System.out.println("Substring: " + str.substring(2, 7));
    System.out.println("Index of Java: " + str.indexOf("Java"));
```

```
        System.out.println("Equals: " + str.trim().equals("Hello Java World"));
        System.out.println("Upper:        "        +        str.toUpperCase());
        System.out.println("Lower:        "        +        str.toLowerCase());
        System.out.println("Trimmed:          "          +          str.trim());
        System.out.println("Replaced: " + str.replace("Java", "Python")); for
        (String word : str.trim().split(" ")) {
            System.out.println("Split word: " + word);
        }
    }
}
```

Component with Controls and Event Handling (MathCalc):

```
import java.awt.*;
import java.awt.event.*;

public class MathCalc extends Frame implements ActionListener {
    TextField tf1, tf2, tf3;
    Button add;

    MathCalc() {
        tf1 = new TextField(); tf2 = new
        TextField(); tf3 = new TextField();
        tf1.setBounds(50,    50,    150,    20);
        tf2.setBounds(50,    80,    150,    20);
        tf3.setBounds(50,    140,    150,    20);
        tf3.setEditable(false);   add   =   new
        Button("+"); add.setBounds(50, 110, 50,
        20);        add.addActionListener(this);
        add(tf1); add(tf2); add(tf3); add(add);
        setSize(300,   300);   setLayout(null);
        setVisible(true);




    }

    public void actionPerformed(ActionEvent e) {
    int a = Integer.parseInt(tf1.getText()); int
    b     =     Integer.parseInt(tf2.getText());
    tf3.setText(String.valueOf(a + b));
    }

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

Draw Line, Rectangle, Oval, Text:

```
import java.awt.*;
```

```
public class DrawGraphics extends Frame {
    public void paint(Graphics g) {
        g.drawLine(50,    50,    100,    100);
        g.drawRect(120,    50,    100,    50);
        g.drawOval(240,    50,    80,    50);
        g.drawString("Hello Java", 100, 200);
    }

    public static void main(String[] args) {
    DrawGraphics f = new DrawGraphics();
    f.setSize(400, 300);
    f.setVisible(true);
    }
}
```

Create a Menu Using Frame:

```
import java.awt.*;
import java.awt.event.*;

public class MenuExample {
    MenuExample() {
        Frame f = new Frame("Menu Example");
        MenuBar mb = new MenuBar(); Menu m1
        = new Menu("File"); MenuItem m11 =
        new MenuItem("Open"); MenuItem m12 =
        new  MenuItem("Exit");  m1.add(m11);
        m1.add(m12);              mb.add(m1);
        f.setMenuBar(mb);      f.setSize(300,
        300); f.setVisible(true);



    }

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

Create a Dialog Box:

```
import java.awt.*;
import java.awt.event.*;

public class DialogBoxExample {
    public static void main(String[] args) {
        Frame f = new Frame("Main Frame");
        Dialog d = new Dialog(f, "Dialog Box Example", true);
        d.setLayout(new FlowLayout());
        d.add(new Label("This is a dialog"));
        Button b = new Button("OK");
        b.addActionListener(e -> d.setVisible(false));
```

```
        d.add(b);
        d.setSize(200, 100);
        f.setSize(300, 300);
        f.setVisible(true);
        d.setVisible(true);
    }
}
```

FlowLayout and BorderLayout:

```
import java.awt.*;

public class LayoutsExample {
    public static void main(String[] args) {
        Frame f = new Frame("Layouts");
        f.setLayout(new  FlowLayout());
        for (int i = 1; i <= 5; i++) {
            f.add(new Button("Button " + i));
        }
        f . s e t S i z e ( 3 0 0 ,  2 0 0 ) ;
        f . s e t V i s i b l e ( t r u e ) ;

        Frame    f2    =    new    Frame("BorderLayout");
        f2.setLayout(new    BorderLayout());    f2.add(new
        Button("North"),  BorderLayout.NORTH);  f2.add(new
        Button("South"),  BorderLayout.SOUTH);  f2.add(new
        Button("East"),   BorderLayout.EAST);   f2.add(new
        Button("West"),   BorderLayout.WEST);   f2.add(new
        Button("Center"),            BorderLayout.CENTER);
        f2.setSize(300, 200); f2.setVisible(true);

    }
}
```

GridLayout and CardLayout:

```
import java.awt.*;

public class LayoutsDemo {
    public static void main(String[] args) {
        Frame f = new Frame("GridLayout");
        f.setLayout(new GridLayout(2, 2));
        for (int i = 1; i <= 4; i++) {
            f.add(new Button("Btn " + i));
        }
        f . s e t S i z e ( 3 0 0 ,  2 0 0 ) ;
        f . s e t V i s i b l e ( t r u e ) ;

        Frame cardFrame = new Frame("CardLayout");
        CardLayout cl = new CardLayout();
        cardFrame.setLayout(cl);
        Button b1 = new Button("Card1");
```

```
        Button b2 = new Button("Card2");
        cardFrame.add("c1",          b1);
        cardFrame.add("c2",          b2);
        cl.show(cardFrame,         "c1");
        cardFrame.setSize(300,      200);
        cardFrame.setVisible(true);
    }
}
```

Frame Displaying Student Information:

```
import java.awt.*;

public class StudentFrame {
    StudentFrame() {
        Frame f = new Frame("Student Info");
        Label l1 = new Label("Name: John");
        Label l2 = new Label("Roll No: 123");
        Label l3 = new Label("Course: BCA");
        l1.setBounds(50,    50,    200,    20);
        l2.setBounds(50,    80,    200,    20);
        l3.setBounds(50,   110,    200,    20);
        f.add(l1);    f.add(l2);    f.add(l3);
        f.setSize(300,                  200);
        f.setLayout(null);
        f.setVisible(true);
    }

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