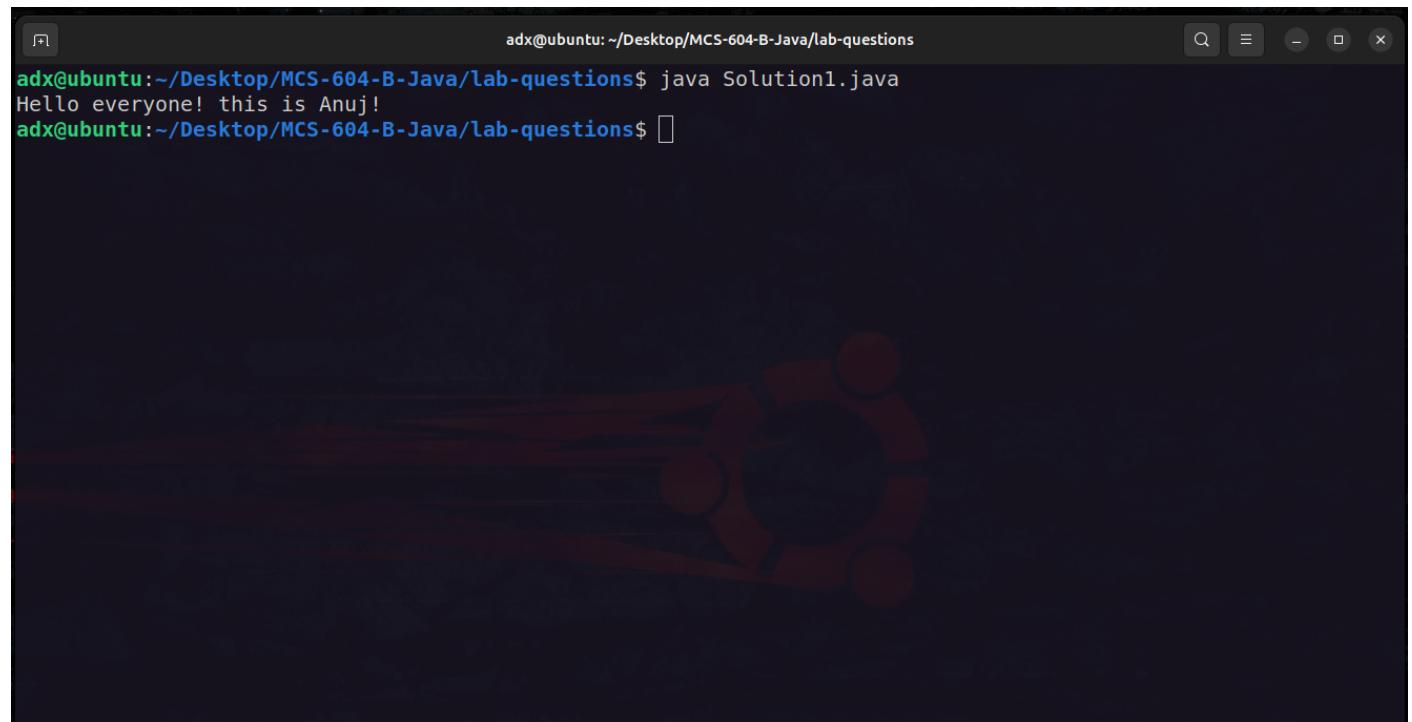


1. Write a program to print a message.

```
class Solution1 {  
    public static void main(String []args) {  
        System.out.println("Hello everyone! this is Anuj!");  
    }  
}
```

Output:

A screenshot of a terminal window on an Ubuntu desktop. The window title bar says "adx@ubuntu: ~/Desktop/MCS-604-B-Java/lab-questions". The terminal itself has a dark background with a faint red Ubuntu logo watermark. The text inside the terminal is white. It shows the command "java Solution1.java" being run, followed by the output "Hello everyone! this is Anuj!". The prompt "adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions\$" is visible at the bottom.

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution1.java  
Hello everyone! this is Anuj!  
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$
```

2. Write a program to find the sum, average, min and max of the 'n' numbers using user input from CLI.

```
class Solution2 {  
    public static void main(String ...args) {  
        int sum = 0, min = Integer.parseInt(args[0]),  
        max = Integer.parseInt(args[0]), n = args.length;  
  
        System.out.print("Given 'n' numbers: ");  
        for(String num : args) {  
            System.out.print(num + " ");  
            int x = Integer.parseInt(num);  
            sum += x;  
            if(x < min) {  
                min = x;  
            }  
            if(x > max) {  
                max = x;  
            }  
        }  
        System.out.println("\n\nMax Value: " + max);  
        System.out.println("Min Value: " + min);  
        System.out.println("The Sum of Integers: " + sum);  
        System.out.println("The Average of Integers: " + (float)sum/n);  
    }  
}
```

Output:

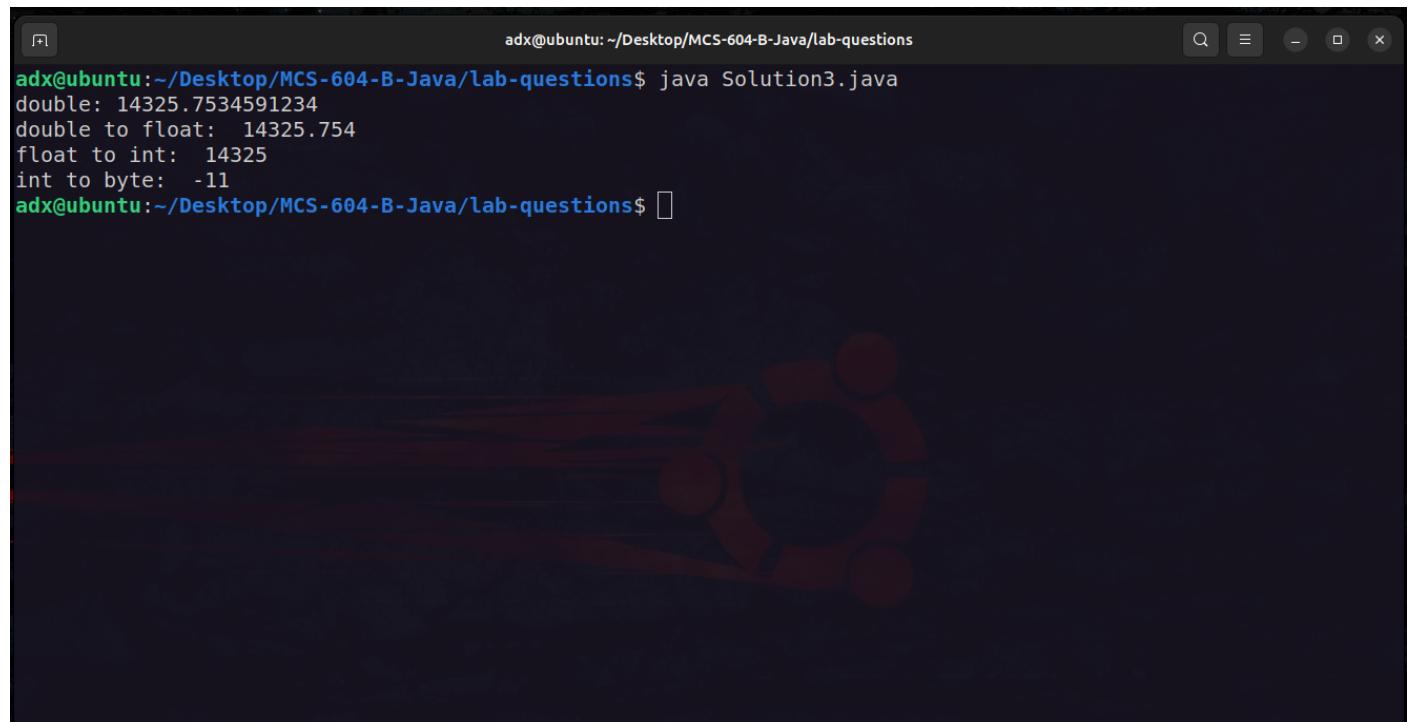


```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution2.java 21 32 65 45 66 25 20 10  
Given 'n' numbers: 21 32 65 45 66 25 20 10  
Max Value: 66  
Min Value: 10  
The Sum of Integers: 284  
The Average of Integers: 35.5  
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ 
```

3. Write a program to demonstrate type casting.

```
class Solution3 {  
    public static void main(String []args) {  
        double d = 14325.7534591234;  
        System.out.println("double: " + d);  
  
        float ff = (float)d;  
        System.out.println("double to float: " + ff);  
  
        int n = (int)ff;  
        System.out.println("float to int: " + n);  
  
        byte b = (byte)n;  
        System.out.println("int to byte: " + b);  
    }  
}
```

Output:



A screenshot of a terminal window titled 'adx@ubuntu: ~/Desktop/MCS-604-B-Java/lab-questions'. The window shows the execution of the 'Solution3.java' program. The output is as follows:

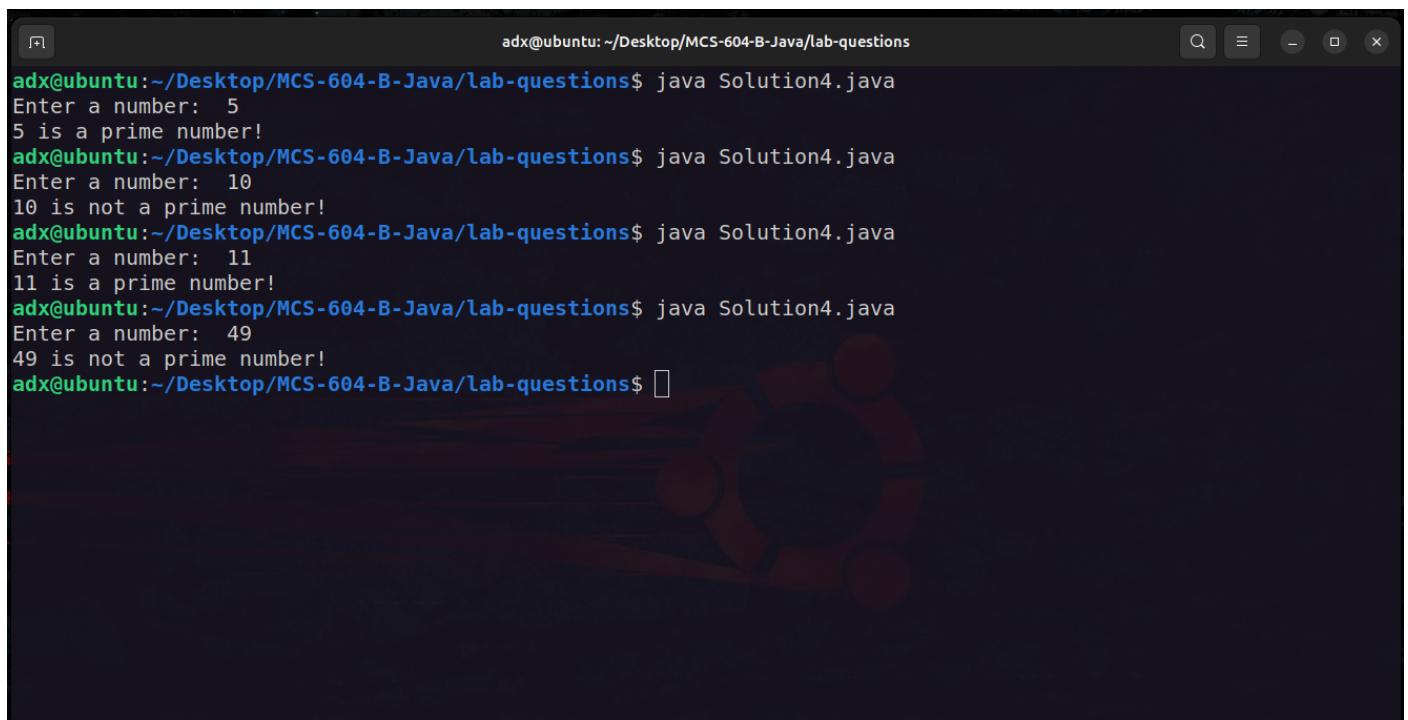
```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution3.java  
double: 14325.7534591234  
double to float: 14325.754  
float to int: 14325  
int to byte: -11  
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$
```

The terminal has a dark background with the Ubuntu logo watermark. The window title bar and the command prompt are in white text. The output text is also in white.

4. Write a program to check whether the given number is prime or not.

```
import java.util.Scanner;
class Solution4 {
    static void checkPrime(int n) {
        if(n == 1) {
            System.out.println(n + " is neither prime nor composite!");
            return;
        }
        for(int i = 2; i < (int) n/2; i++) {
            if(n % i == 0) {
                System.out.println(n + " is not a prime number!");
                return;
            }
        }
        System.out.println(n + " is a prime number!");
    }
    public static void main(String []args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a number: ");
        int num = sc.nextInt();
        checkPrime(num);
        sc.close();
    }
}
```

Output:



A screenshot of a terminal window titled 'adx@ubuntu: ~/Desktop/MCS-604-B-Java/lab-questions'. The window shows the execution of a Java program named 'Solution4.java'. The user enters several numbers (5, 10, 11, 49) and the program outputs whether each is prime or not. The terminal has a dark background with a faint red Ubuntu logo watermark.

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution4.java
Enter a number: 5
5 is a prime number!
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution4.java
Enter a number: 10
10 is not a prime number!
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution4.java
Enter a number: 11
11 is a prime number!
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution4.java
Enter a number: 49
49 is not a prime number!
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$
```

5. Write a program to find out the HCF and LCM.

```
import java.util.Scanner;
class Solution5 {
    static int findHCF(int a, int b) {
        while(b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
    static int findLCM(int a, int b) {
        return (a * b) / findHCF(a, b);
    }
    public static void main(String []args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter two numbers: \na >> ");
        int a = sc.nextInt();
        System.out.print("b >> ");
        int b = sc.nextInt();
        int hcf = findHCF(a, b), lcm = findLCM(a, b);
        System.out.printf("\nThe HCF of %d and %d is %d", a, b, hcf);
        System.out.printf("\nThe LCM of %d and %d is %d\n\n", a, b, lcm);
        sc.close();
    }
}
```

Output:

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution5.java
Enter two numbers:
a >> 12
b >> 45

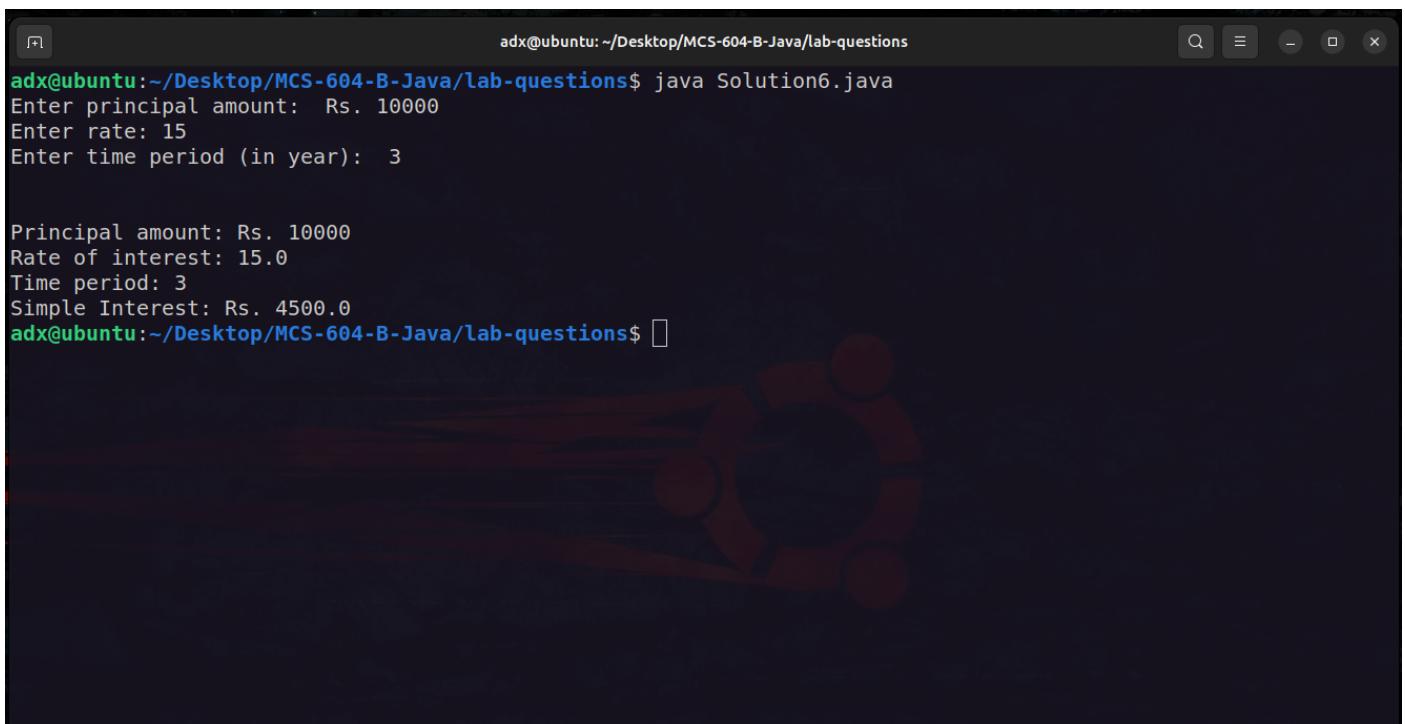
The HCF of 12 and 45 is 3
The LCM of 12 and 45 is 180

adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$
```

6. Write a program to calculate Simple Interest and data is taken as input from users.

```
import java.util.Scanner;
class Solution6 {
    static void calculateSI(int principal, float rate, int time) {
        System.out.println("\n\nPrincipal amount: Rs. " + principal);
        System.out.println("Rate of interest: " + rate);
        System.out.println("Time period: " + time);
        float si = (principal * rate * time) / 100;
        System.out.println("Simple Interest: Rs. " + si);
    }
    public static void main(String []args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter principal amount: Rs. ");
        int p = sc.nextInt();
        System.out.print("Enter rate: ");
        float r = sc.nextFloat();
        System.out.print("Enter time period (in year): ");
        int t = sc.nextInt();
        calculateSI(p, r, t);
        sc.close();
    }
}
```

Output:



A screenshot of a terminal window titled "adx@ubuntu: ~/Desktop/MCS-604-B-Java/lab-questions". The window shows the execution of a Java program named "Solution6.java". The user inputs principal amount (10000), rate (15), and time period (3). The program then outputs the calculated simple interest (4500.0).

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution6.java
Enter principal amount: Rs. 10000
Enter rate: 15
Enter time period (in year): 3

Principal amount: Rs. 10000
Rate of interest: 15.0
Time period: 3
Simple Interest: Rs. 4500.0
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$
```

7. Write a program to create a simple class to find out the Area and Perimeter of Rectangle and Box using super and this keyword.

```
class Rectangle {  
    int l, b;  
    Rectangle(int l, int b) {  
        this.l = l;  
        this.b = b;  
    }  
  
    void calcArea() {  
        System.out.println("Area of the given Rectangle is: " + l*b);  
    }  
  
    void calcPerimeter() {  
        int p = 2*(l+b);  
        System.out.println("Perimeter of the given Rectangle is: " + p);  
    }  
}  
  
class Box extends Rectangle {  
    int h;  
    Box(int l, int b, int h) {  
        super(l, b);  
        this.h = h;  
    }  
  
    void calcVolume() {  
        System.out.println("Volume of the given Box is: " + l*b*h);  
    }  
}  
  
class Solution7 {  
    public static void main(String []args) {  
        Rectangle r = new Rectangle(8, 4);  
        r.calcArea();  
        r.calcPerimeter();  
  
        Box b = new Box(8, 4, 5);  
        b.calcVolume();  
    }  
}
```

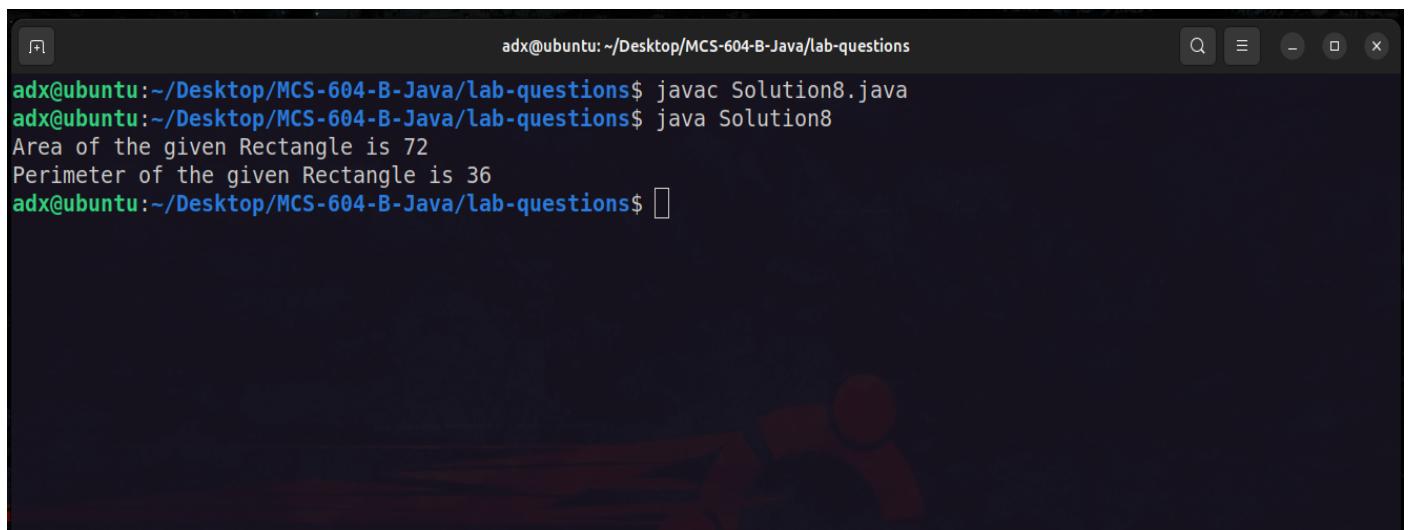
Output:

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ javac Solution7.java
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution7
Area of the given Rectangle is: 32
Perimeter of the given Rectangle is: 24
Volume of the given Box is: 160
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$
```

8. Write a program to design a class Shape to implement runtime polymorphism using abstract methods and classes.

```
abstract class Shape {  
    abstract void calcArea();  
    abstract void calcPerimeter();  
}  
  
class Rectangle extends Shape {  
    int length, width;  
    Rectangle(int length, int width) {  
        this.length = length;  
        this.width = width;  
    }  
    void calcArea() {  
        int area = length * width;  
        System.out.println("Area of the given Rectangle is " + area);  
    }  
    void calcPerimeter() {  
        int p = 2 * (length + width);  
        System.out.println("Perimeter of the given Rectangle is " + p);  
    }  
}  
  
class Solution8 {  
    public static void main(String []args) {  
        Rectangle r = new Rectangle(12, 6);  
        r.calcArea();  
        r.calcPerimeter();  
    }  
}
```

Output:



The screenshot shows a terminal window on a Linux system (Ubuntu). The user has navigated to the directory `~/Desktop/MCS-604-B-Java/lab-questions`. They first run the `javac` command to compile the `Solution8.java` file. After compilation is successful, they run the `java` command to execute the program. The output shows the calculated area (72) and perimeter (36) of a rectangle with dimensions 12 and 6 respectively.

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ javac Solution8.java  
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution8  
Area of the given Rectangle is 72  
Perimeter of the given Rectangle is 36  
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ 
```

9. Write a program to demonstrate the use of different String class methods.

```
class Solution9 {  
    public static void main(String []args) {  
        String msg = "This is some sample string with some words";  
        System.out.println("Original String: " + msg + "\n");  
  
        int len = msg.length();  
        System.out.println("Total Number of Characters: " + len);  
  
        String uppr = msg.toUpperCase();  
        System.out.println("Converted to Uppercase: " + uppr);  
  
        String lowr = msg.toLowerCase();  
        System.out.println("Converted to Lowercase: " + lowr);  
  
        int idxs = msg.indexOf("some");  
        System.out.println("Index of 'some': " + idxs);  
  
        String subs1 = msg.substring(8), subs2 = msg.substring(8, 19);  
        System.out.println("Substring from index-8: " + subs1);  
        System.out.println("Substring from index 8 to 19: " + subs2);  
  
        String ms = "      hello world!      ";  
        System.out.println("Original String: " + ms);  
  
        ms = ms.trim();  
        System.out.println("String with trimmed spaces: " + ms);  
  
        ms = ms.replace("world", "universe");  
        System.out.println("Replacing 'world' with 'universe': " + ms);  
  
        StringBuffer rev = new StringBuffer(ms).reverse();  
        System.out.println("Reversing: " + ms + " -> " + rev);  
    }  
}
```

Output:

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution9.java
Original Stirng: This is some sample string with some words

Total Number of Characters: 42
Converted to Uppercase: THIS IS SOME SAMPLE STRING WITH SOME WORDS
Converted to Lowercase: this is some sample string with some words
Index of 'some': 8
Substring from index-8: some sample string with some words
Substring from index 8 to 19: some sample
Original String: hello world!
String with trimmed spaces: hello world!
Replacing 'world' with 'universe': hello universe!
Reversing: hello universe! -> !esrevinu olleh
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$
```

10. Write a program to handle multiple Exceptions.

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

class Solution10 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int x1 = 0, y1 = 0;
        try {
            System.out.print("Enter the value of X:   ");
            int x = s.nextInt();
            System.out.print("Enter the value of Y:   ");
            int y = s.nextInt();
            x1 = x;
            y1 = y;
            float div = x / y;
            System.out.println("The value of "+x+" / "+y+" is "+div);
        }
        catch(InputMismatchException nfe) {
            System.out.println("Values of X & Y must be an integer!");
            return;
        }
        catch(ArithmeticException ex) {
            System.out.println("Can not Divided by zero!");
        }
        catch(Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
        finally {
            s.close();
        }
        int sum = x1 + y1;
        int sub = x1 - y1;
        int mul = x1 * y1;

        System.out.println("The value of "+x1+" + "+y1+" is "+sum);
        System.out.println("The value of "+x1+" - "+y1+" is "+sub);
        System.out.println("The value of "+x1+" * "+y1+" is "+mul);
    }
}
```

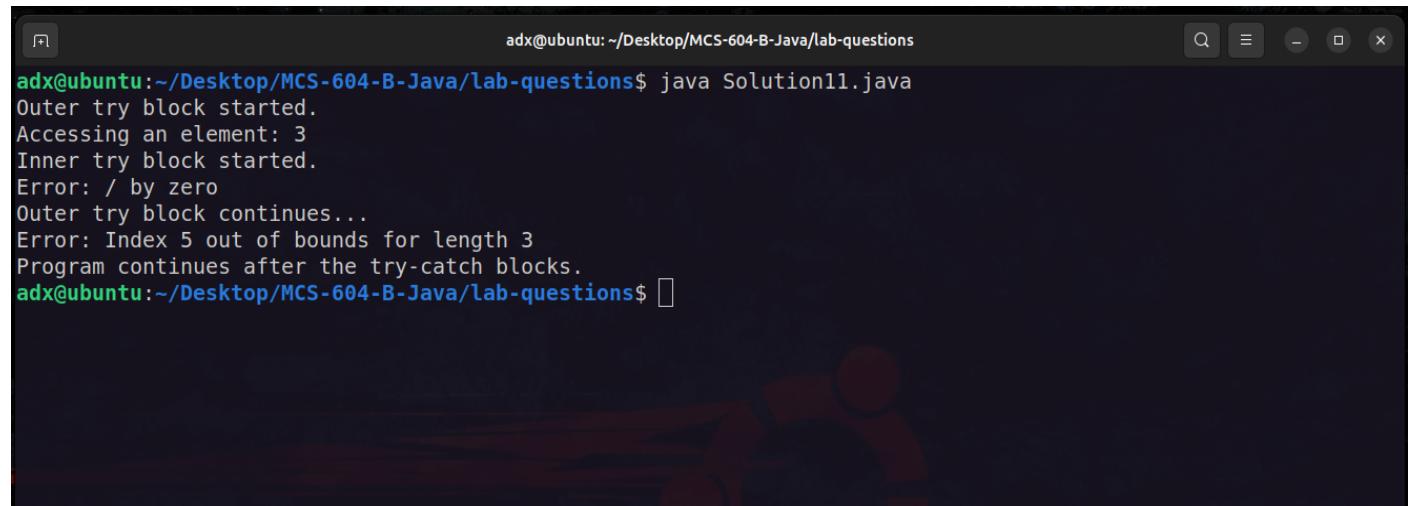
Output:

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution10.java
Enter the value of X: 5
Enter the value of Y: "0"
InputMismatchException Occurred: Value of both X & Y must be an integer!!
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution10.java
Enter the value of X: 5
Enter the value of Y: 0
ArithmaticException Occurred: Can not Divided by zero!
The value of 5 + 0 is 5
The value of 5 - 0 is 5
The value of 5 * 0 is 0
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$
```

11. Write a program to implement nested try-catch blocks to handle Exception.

```
class Solution11 {
    public static void main(String[] args) {
        try {
            // Outer try block
            System.out.println("Outer try block started.");
            int[] num = {1, 2, 3};
            System.out.println("Accessing an element: " + num[2]);
            try {
                // Inner try block
                System.out.println("Inner try block started.");
                int result = 10 / 0;
                System.out.println("Result: " + result);
            }
            catch (ArithmaticException e) {
                System.out.println("Error: " + e.getMessage());
            }
            System.out.println("Outer try block continues...");
            System.out.println("Accessing an invalid index: " + num[5]);
        }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: " + e.getMessage());
        }
        System.out.println("Continues after the try-catch blocks.");
    }
}
```

Output:

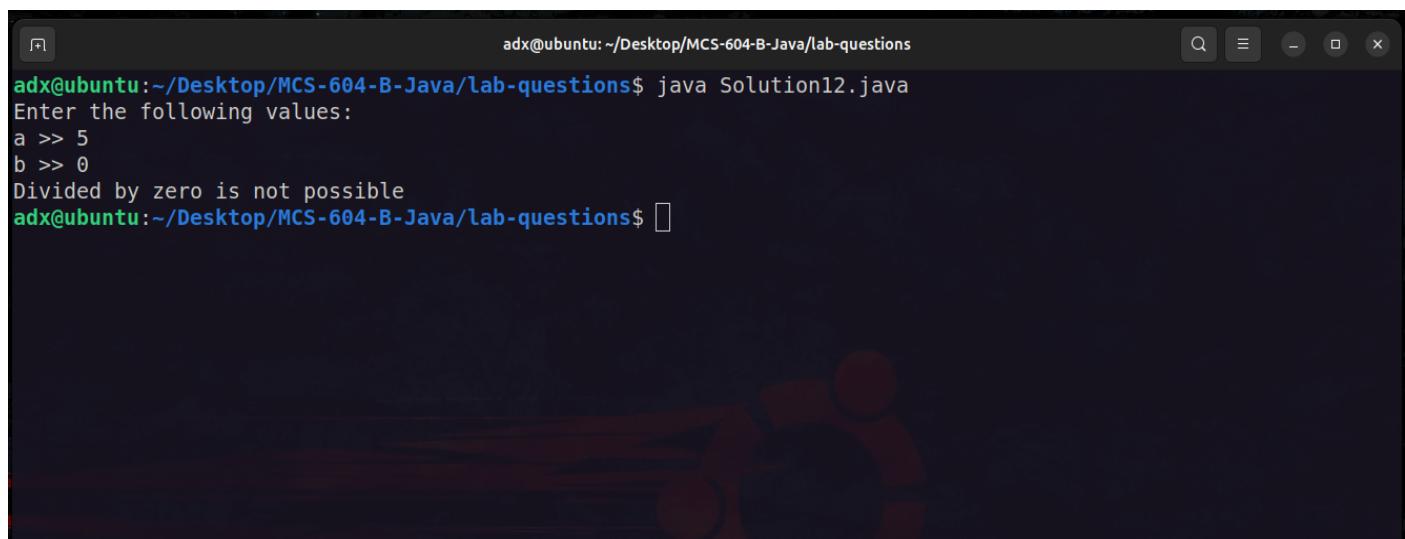


```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution11.java
Outer try block started.
Accessing an element: 3
Inner try block started.
Error: / by zero
Outer try block continues...
Error: Index 5 out of bounds for length 3
Program continues after the try-catch blocks.
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$
```

12. Write a program that implements throw and throws.

```
import java.util.Scanner;
class Solution12 {
    static void divide(int a, int b) throws ArithmeticException {
        if(b == 0) {
            throw new ArithmeticException("Divided by zero not possible");
        }
        else {
            float res = (float) a / b;
            System.out.println(a + "/" + b + " = " + res);
        }
    }
    public static void main(String[] args) {
        try {
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter the following values: \na >> ");
            int a = sc.nextInt();
            System.out.print("b >> ");
            int b = sc.nextInt();
            divide(a, b);
        }
        catch(ArithmeticException ae) {
            System.out.println(ae.getMessage());
        }
    }
}
```

Output:



```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution12.java
Enter the following values:
a >> 5
b >> 0
Divided by zero is not possible
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ 
```

13. Write a program to implement custom Exceptions.

```
import java.util.Scanner;

class NegativeRadiusException extends Exception {
    @Override
    public String getMessage() {
        return "Radius can not be Negative!";
    }

    @Override
    public String toString() {
        return "Radius can not be Negative!";
    }
}

class Solution13 {
    static void calcCircleArea(int rad) throws NegativeRadiusException {
        if(rad < 0) {
            throw new NegativeRadiusException();
        }
        else {
            double area = Math.PI * rad * rad;
            System.out.println("The Area of the Circle is " + area);
        }
    }

    public static void main(String[] args) {
        System.out.print("Enter the Radius of the Circle:   ");
        int radius = new Scanner(System.in).nextInt();
        try {
            calcCircleArea(radius);
        }
        catch(NegativeRadiusException nre) {
            System.out.println(nre.getMessage());
        }
    }
}
```

Output:

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ javac Solution13.java
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution13
Enter the Radius of the Circle: -5
Radius can not be Negative!
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution13
Enter the Radius of the Circle: 5
The Area of the Circle is 78.53981633974483
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$
```

14. Write a program to implement the concept of multiple interfaces.

```
import java.util.Scanner;

interface Area {
    public void calcArea();
}

interface Perimeter {
    public void calcPerimeter();
}

class Circle implements Area, Perimeter {
    int radius;
    Circle(int radius) {
        this.radius = radius;
    }
    public void calcArea() {
        double area = Math.PI * radius * radius;
        System.out.println("The Area of the Circle is "+ area);
    }
    public void calcPerimeter() {
        double peri = 2 * Math.PI * radius;
        System.out.println("The Circumference of the Circle is "+ peri);
    }
}

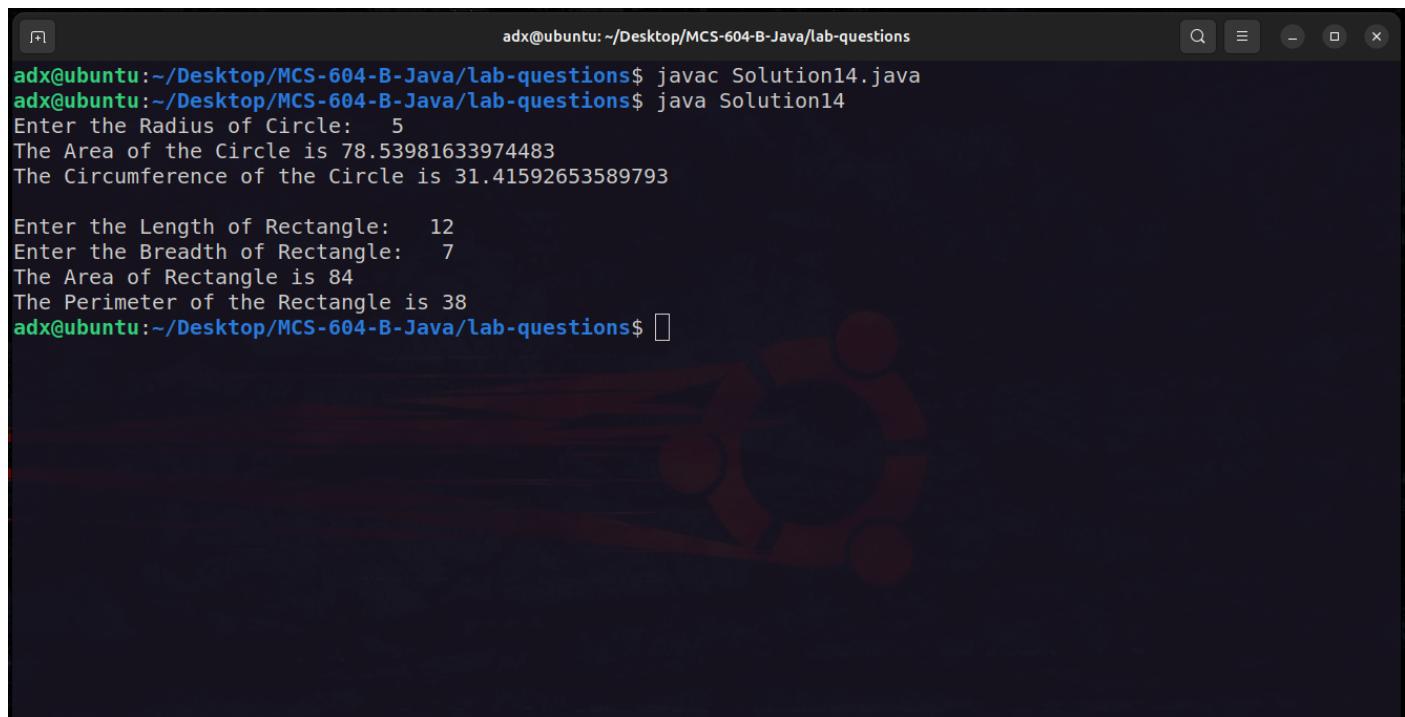
class Rectangle implements Area, Perimeter {
    int l, b;
    Rectangle(int length, int breadth) {
        l = length;
        b = breadth;
    }
    public void calcArea() {
        int area = l*b;
        System.out.println("The Area of Rectangle is "+ area);
    }
    public void calcPerimeter() {
        int peri = 2*(l+b);
        System.out.println("The Perimeter of the Rectangle is "+peri);
    }
}
```

```

class Solution14 {
    public static void main(String[ ] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the Radius of Circle:   ");
        int radius = sc.nextInt();
        Circle cc = new Circle(radius);
        cc.calcArea();
        cc.calcPerimeter();
        System.out.println("");
        System.out.print("Enter the Length of Rectangle:   ");
        int length = sc.nextInt();
        System.out.print("Enter the Breadth of Rectangle:   ");
        int breadth = sc.nextInt();
        Rectangle rr = new Rectangle(length, breadth);
        rr.calcArea();
        rr.calcPerimeter();
        sc.close();
    }
}

```

Output:



The screenshot shows a terminal window on an Ubuntu desktop environment. The title bar indicates the session is running on 'adx@ubuntu'. The terminal window displays the following text:

```

adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ javac Solution14.java
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution14
Enter the Radius of Circle: 5
The Area of the Circle is 78.53981633974483
The Circumference of the Circle is 31.41592653589793

Enter the Length of Rectangle: 12
Enter the Breadth of Rectangle: 7
The Area of Rectangle is 84
The Perimeter of the Rectangle is 38
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ 

```

The terminal window has a dark background with light-colored text. The Ubuntu logo is visible in the bottom right corner of the desktop area.

15. Write a program to design a class account using the inheritance and static that show all functions of the bank(withdrawal, deposit) and generate account numbers dynamically.

```
class Bank {  
  
    static int account_number_generator = 10000;  
    String name;  
    int acc_no;  
    float balance;  
  
    Bank(String name, float min_deposit_amount) {  
        this.name = name;  
        this.acc_no = account_number_generator;  
        this.balance = min_deposit_amount;  
        System.out.printf("\n\nHello %s, your account number is: %d", name, acc_no);  
        account_number_generator++;  
    }  
  
    void display() {  
        System.out.println("\n\nWelcome " + name + ", to your account!");  
        System.out.println("Account Number: " + acc_no);  
        System.out.println("Current Balance: Rs." + balance);  
    }  
  
    void getBalance() {  
        System.out.println("Your current balance: Rs." + balance);  
    }  
  
    void withdraw(float amt) {  
        if(balance <= amt || balance == 1000) {  
            System.out.println("Sorry! you can't withdraw money!");  
        }  
        else {  
            balance -= amt;  
            System.out.println("\nAmount withdrawn: Rs." + amt);  
            getBalance();  
        }  
    }  
  
    void deposit(float amt) {  
        if(amt == 0.0) {  
            System.out.println("Sorry! you can't deposit Rs. 0.0");  
        }  
        else {  
            balance += amt;  
            System.out.println("\nAmount deposited: Rs." + amt);  
            getBalance();  
        }  
    }  
}
```

```
class Solution15 {  
    public static void main(String[] args) {  
        Bank user1 = new Bank("John", 5055.5f);  
        user1.display();  
        user1.deposit(2550.5f);  
        user1.withdraw(800);  
  
        Bank user2 = new Bank("Emily", 2015.6f);  
        user2.display();  
        user2.deposit(1550.5f);  
        user2.withdraw(500);  
    }  
}
```

Output:

```
adx@ubuntu: ~/Desktop/MCS-604-B-Java/lab-questions  
  
Hello John, your account number is: 10000  
  
Welcome John, to your account!  
Account Number: 10000  
Current Balance: Rs.5055.5  
  
Amount deposited: Rs.2550.5  
Your current balance: Rs.7606.0  
  
Amount withdrawn: Rs.800.0  
Your current balance: Rs.6806.0  
  
  
Hello Emily, your account number is: 10001  
  
Welcome Emily, to your account!  
Account Number: 10001  
Current Balance: Rs.2015.6  
  
Amount deposited: Rs.1550.5  
Your current balance: Rs.3566.1  
  
Amount withdrawn: Rs.500.0  
Your current balance: Rs.3066.1  
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$
```

16. Write a program to create a package that accesses the member of the external class as well as the same package.

The screenshot shows the Visual Studio Code interface with two files open:

- Main.java** (active tab):

```
import java.util.Scanner;
import search.unsorted.LinearSearch;

public class Main {
    static void display(int arr[], String caption) {
        System.out.print(caption);
        for(int i=0; i < arr.length; i++) {
            System.out.print(arr[i]+ " ");
        }
        System.out.println("");
    }

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

        System.out.print("Enter the size of array: ");
        int n = sc.nextInt();

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

        display(arr, caption:"Given array: ");
        System.out.print("Enter the number to be searched: ");
        int target = sc.nextInt();

        ls.search(arr, target);
        sc.close();
    }
}
```
- LinearSearch.java**:

```
package search.unsorted;

public class LinearSearch {
    public void search(int[] arr, int x) {
        for(int i=0; i < arr.length; i++) {
            if(arr[i]==x) {
                System.out.println(x + " found at index " + i);
                return;
            }
        }
        System.out.println(x + " not found in this array!");
    }
}
```

The Explorer sidebar shows a package named "CUSTOM_PACKAGE" containing "search/unsorted" which includes "LinearSearch.class" and "Main.class". The status bar at the bottom right indicates "Ln 37, Col 1 Spaces: 4 UTF-8 LF Java Go Live".

Location: custom_package/search/unsorted/LinearSearch.java

```
package search.unsorted;

public class LinearSearch {

    public void search(int[] arr, int x) {
        for(int i=0; i < arr.length; i++) {
            if(arr[i]==x) {
                System.out.println(x + " found at index " + i);
                return;
            }
        }
        System.out.println(x + " not found in this array!");
    }
}
```

Location: custom_package/Main.java

```
import java.util.Scanner;
import search.unsorted.LinearSearch;

public class Main {

    static void display(int arr[], String caption) {
        System.out.print(caption);
        for(int i=0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println("");
    }

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

        System.out.print("Enter the size of array: ");
        int n = sc.nextInt();

        int arr[] = new int[n];

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

        display(arr, "Given array: ");

        System.out.print("Enter the number to be searched: ");
        int target = sc.nextInt();

        ls.search(arr, target);
        sc.close();
    }
}
```

Output:

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/custom_package$ ls
Main.java  search
adx@ubuntu:~/Desktop/MCS-604-B-Java/custom_package$ javac Main.java
adx@ubuntu:~/Desktop/MCS-604-B-Java/custom_package$ java Main
Enter the size of array: 7
Enter 7 numbers!
12 32 65 52 99 46 35
Given array: 12 32 65 52 99 46 35
Enter the number to be searched: 52
52 found at index 3
adx@ubuntu:~/Desktop/MCS-604-B-Java/custom_package$
```

17. Write a program that shows the partial implementation of Interface (Calculation of Salary of Employee).

```
interface SalaryCalculator {
    double calculateSalary();
    String getDetails();           // Abstract method
}

abstract class Employee implements SalaryCalculator {
    String name;
    private double baseSalary;
    private double bonus;

    public Employee(String name, double baseSalary, double bonus) {
        this.name = name;
        this.baseSalary = baseSalary;
        this.bonus = bonus;
    }
    @Override
    public double calculateSalary() {
        return baseSalary + bonus;
    }
    // getDetails() method remains abstract in this class
}

class Manager extends Employee {
    public Manager(String name, double baseSalary, double bonus) {
        super(name, baseSalary, bonus);
    }
    @Override
    public String getDetails() {
        return "Manager Name: " + name + ", Salary: $" + calculateSalary();
    }
}

public class Solution17 {
    public static void main(String[] args) {
        Employee manager = new Manager("John Doe", 50000, 5000);
        System.out.println(manager.getDetails());
    }
}
```

Output:

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ javac Solution17.java
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ java Solution17
Manager Name: John Doe, Salary: $55000.0
adx@ubuntu:~/Desktop/MCS-604-B-Java/lab-questions$ 
```

18. Write a program to create an Arithmetic Math Calculator using Java Swing and AWT Event Handling.

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

class SwingGUI extends JFrame implements ActionListener {
    JTextField t1, t2;
    JLabel lb1, lb2, lb3;
    JButton sum_btn, sub_btn, mul_btn, div_btn;

    public SwingGUI() {
        super("BASIC CALCULATOR");
        lb1 = new JLabel("Enter First Number:   ");
        lb1.setBounds(90, 20, 150, 30);
        add(lb1);

        t1 = new JTextField(30);
        t1.setBounds(90, 50, 150, 30);
        add(t1);

        lb2 = new JLabel("Enter Second Number:   ");
        lb2.setBounds(90, 80, 150, 30);
        add(lb2);

        t2 = new JTextField(30);
        t2.setBounds(90, 110, 150, 30);
        add(t2);

        lb3 = new JLabel("Result:   ");
        lb3.setBounds(90, 160, 250, 30);
        add(lb3);

        sum_btn = new JButton(" + ");
        sum_btn.setBounds(90, 200, 50, 30);
        add(sum_btn);
        sum_btn.addActionListener(this);

        sub_btn = new JButton(" - ");
        sub_btn.setBounds(160, 200, 50, 30);
        add(sub_btn);
        sub_btn.addActionListener(this);

        mul_btn = new JButton(" * ");
        mul_btn.setBounds(230, 200, 50, 30);
        add(mul_btn);
        mul_btn.addActionListener(this);

        div_btn = new JButton(" / ");
        div_btn.setBounds(300, 200, 50, 30);
        add(div_btn);
        div_btn.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == sum_btn) {
            int num1 = Integer.parseInt(t1.getText());
            int num2 = Integer.parseInt(t2.getText());
            int result = num1 + num2;
            lb3.setText("Result: " + result);
        }
        else if (e.getSource() == sub_btn) {
            int num1 = Integer.parseInt(t1.getText());
            int num2 = Integer.parseInt(t2.getText());
            int result = num1 - num2;
            lb3.setText("Result: " + result);
        }
        else if (e.getSource() == mul_btn) {
            int num1 = Integer.parseInt(t1.getText());
            int num2 = Integer.parseInt(t2.getText());
            int result = num1 * num2;
            lb3.setText("Result: " + result);
        }
        else if (e.getSource() == div_btn) {
            int num1 = Integer.parseInt(t1.getText());
            int num2 = Integer.parseInt(t2.getText());
            int result = num1 / num2;
            lb3.setText("Result: " + result);
        }
    }
}
```

```

mul_btn = new JButton(" * ");
mul_btn.setBounds(230, 200, 50, 30);
add(mul_btn);
mul_btn.addActionListener(this);

div_btn = new JButton(" / ");
div_btn.setBounds(300, 200, 50, 30);
add(div_btn);
div_btn.addActionListener(this);

setLayout(null);
setSize(600, 400);
setVisible(true);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

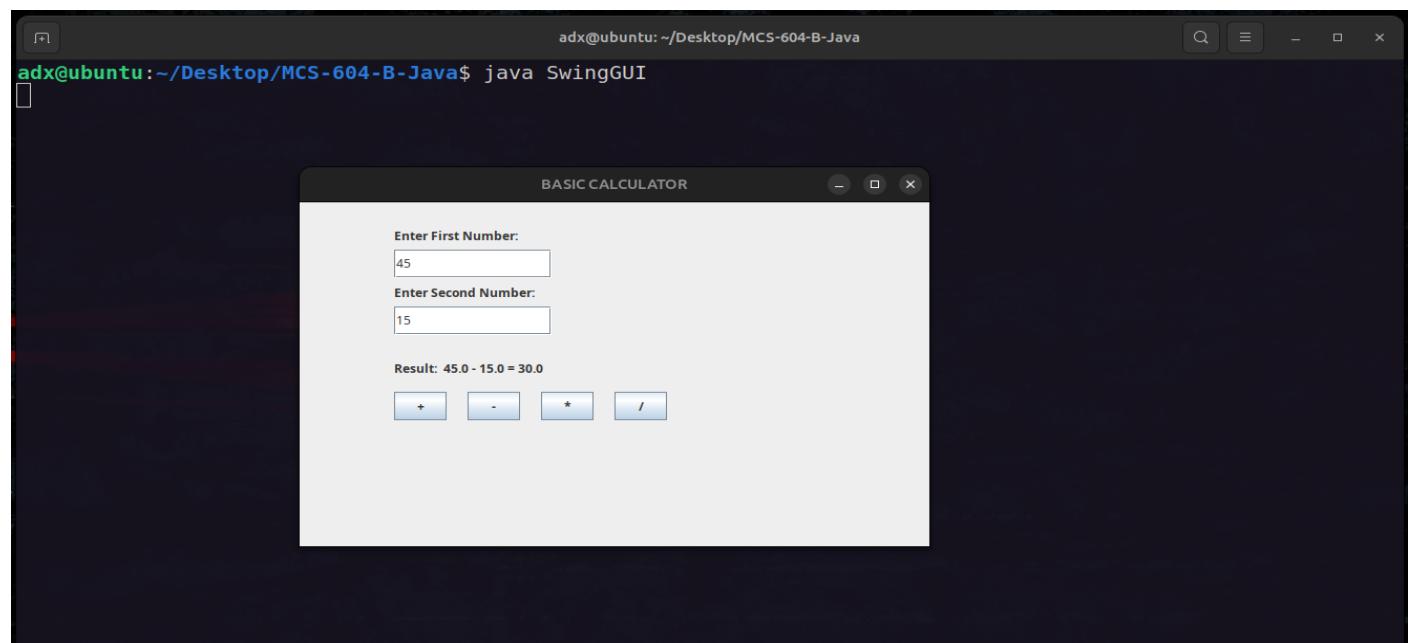
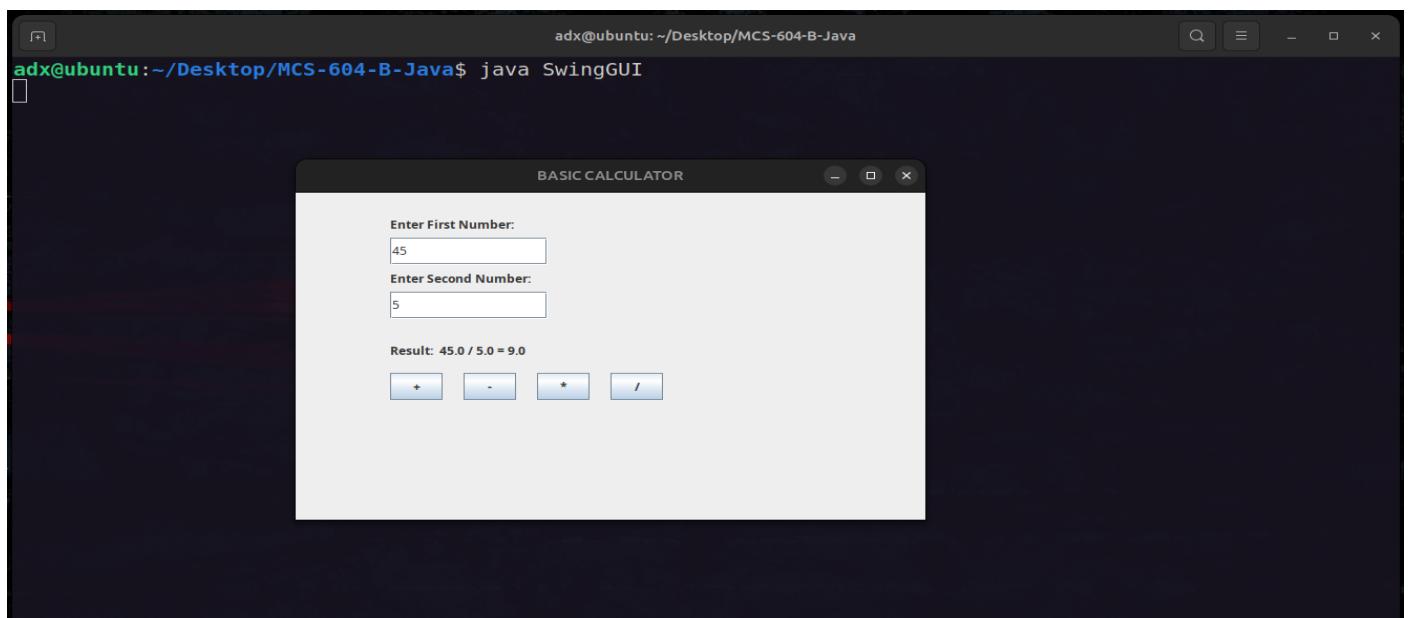
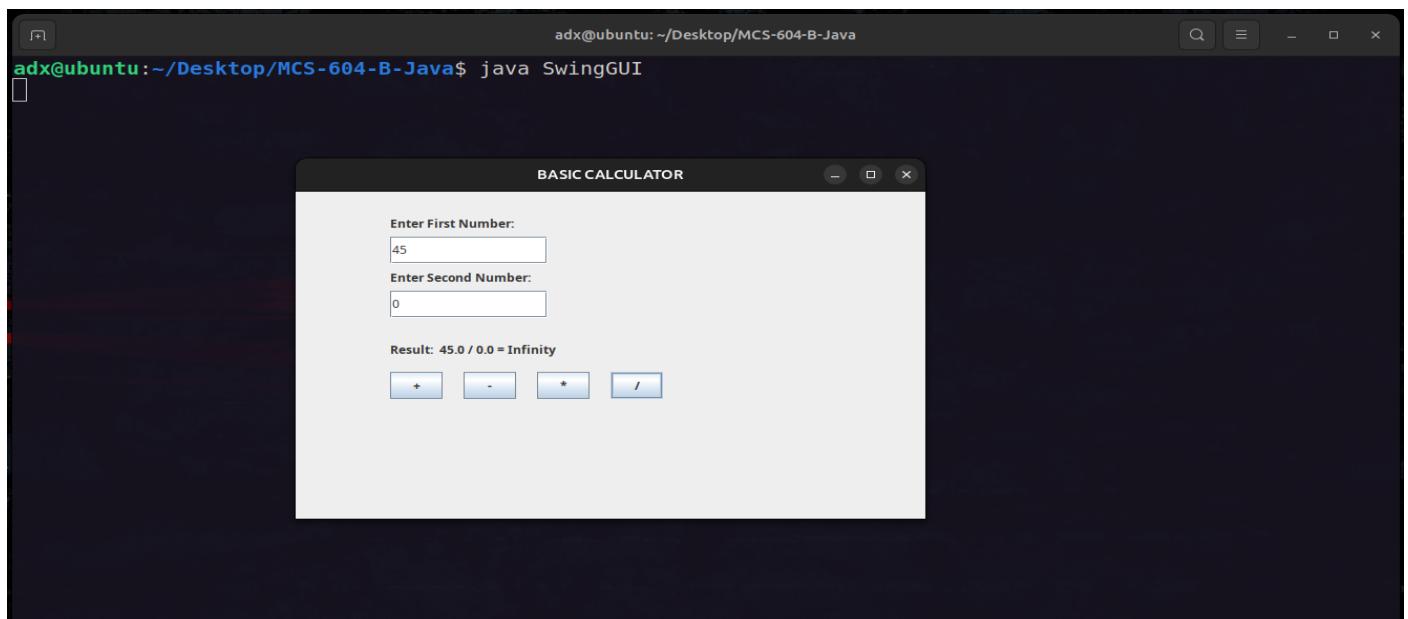
@Override
public void actionPerformed(ActionEvent e) {
    float a = Float.parseFloat(t1.getText());
    float b = Float.parseFloat(t2.getText());

    if (e.getSource().equals(sum_btn)) {
        float sum = a + b;
        lb3.setText("Result: " + a + " + " + b + " = " + String.valueOf(sum));
    }
    else if (e.getSource().equals(sub_btn)) {
        float sub = a - b;
        lb3.setText("Result: " + a + " - " + b + " = " + String.valueOf(sub));
    }
    else if (e.getSource().equals(mul_btn)) {
        float mul = a * b;
        lb3.setText("Result: " + a + " * " + b + " = " + String.valueOf(mul));
    }
    else if (e.getSource().equals(div_btn)) {
        double div = a / (b * 1.0);
        lb3.setText("Result: " + a + " / " + b + " = " + String.valueOf(div));
    }
}

public static void main(String[] args) {
    SwingGUI gui = new SwingGUI();
}
}

```

Output:

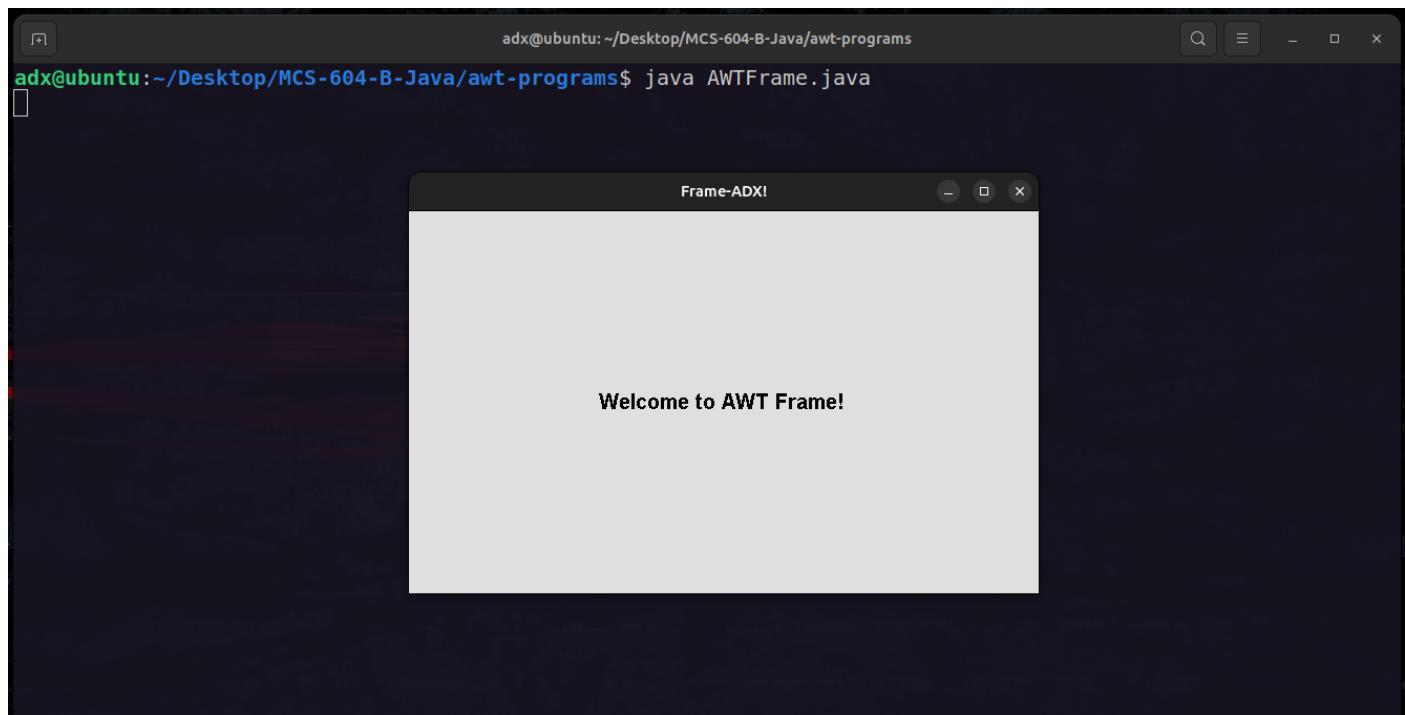


19. Write a program to create a frame window using Frame class. (AWT)

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

class AWTFrame {
    public static void main(String[] args) {
        Frame frame = new Frame("Frame-ADX!");
        frame.setSize(600, 400);
        frame.setLayout(new GridBagLayout());
        Label lb = new Label("Welcome to AWT Frame!");
        lb.setFont(new Font("Arial", Font.BOLD, 20));
        frame.add(lb);
        frame.setVisible(true);
        frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                frame.dispose();
                System.exit(0);
            }
        });
    }
}
```

Output:



20. Write a program to create UI components on frame windows using Frame class. (AWT)

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

class UIComponent {
    public static void main(String[] args) {
        Frame frame = new Frame("Simple AWT UI");
        frame.setSize(400, 300);
        frame.setLayout(null);

        // Add a Label
        Label label = new Label("Enter your name:");
        label.setBounds(50, 50, 120, 30);
        frame.add(label);

        // Add a TextField
        TextField textField = new TextField();
        textField.setBounds(180, 50, 150, 30);
        frame.add(textField);

        // Add a Button
        Button button = new Button("Submit");
        button.setBounds(150, 100, 80, 30);
        frame.add(button);

        // Add an Action Listener for the Button
        button.addActionListener(e -> {
            String name = textField.getText();
            System.out.println("Hello, " + name + "!");
        });

        // Add Window Listener to close the frame
        frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                frame.dispose();
                System.exit(0);
            }
        });

        // Make the frame visible
        frame.setVisible(true);
    }
}
```

Output:

```
adx@ubuntu:~/Desktop/MCS-604-B-Java/awt-programs$ java UIComponent.java
Hello, Anuj!

```

The screenshot shows a terminal window and a graphical application window side-by-side. The terminal window at the top has a dark background and contains the command `java UIComponent.java` and its output, "Hello, Anuj!". Below the terminal is a graphical application titled "Simple AWT UI". This window has a light gray background and contains a text input field with the placeholder "Enter your name:" and the value "Anuj", followed by a "Submit" button.

21. Write a program to implement ListBox. (AWT)

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

public class ListBox {
    public static void main(String[] args) {
        Frame frame = new Frame("Frame-ListBox");
        frame.setSize(600, 400);
        frame.setLayout(null);
        // Create a Label
        Label label = new Label("Select item(s):");
        label.setFont(new Font("Arial", Font.BOLD, 16));
        label.setBounds(50, 70, 250, 30);
        frame.add(label);

        List listBox = new List(6, true);
        listBox.setBounds(50, 100, 150, 150);
        listBox.add("Potato");
        listBox.add("Tomato");
        listBox.add("Apple");
        listBox.add("Pineapple");
        listBox.add("Orange");
        listBox.add("Mango");
        frame.add(listBox);
        // Create a Button to show selected items
        Button button = new Button("Show Selections");
        button.setBounds(220, 100, 150, 30);
        frame.add(button);
        // Create a Label to display the selected items
        Label resultLabel = new Label();
        resultLabel.setBounds(220, 150, 350, 30);
        resultLabel.setFont(new Font("Arial", Font.BOLD, 16));
        frame.add(resultLabel);
        // Add an Action Listener to the Button
        button.addActionListener(e -> {
            String[] selectedItems = listBox.getSelectedItems();
            if (selectedItems.length > 0) {
                String s = "You have selected: ";
                StringBuilder result = new StringBuilder(s);
                for (String item : selectedItems) {
                    result.append(item).append(", ");
                }
            }
        });
    }
}
```

```

        // Remove the trailing comma and space
        result.setLength(result.length() - 2);
        resultLabel.setText(result.toString());
        System.out.println(result.toString());
    }
    else {
        resultLabel.setText("No item selected");
        System.out.println("No item selected");
    }
}) ;

// Add a Window Listener to handle window close events
frame.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        frame.dispose();
        System.exit(0);
    }
}) ;

frame.setVisible(true);
}
}

```

Output:



22. Write a program to implement Choice, Checkbox, RadioButton with AWT event handling. (AWT)

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

public class AWTComponents {
    public static void main(String[] args) {
        Frame frame = new Frame("AWT Components - ADx");
        frame.setSize(500, 400);
        frame.setLayout(null);

        // ----- Choice (Dropdown) -----
        Label choiceLabel = new Label("Select a fruit:");
        choiceLabel.setFont(new Font("Arial", Font.BOLD, 14));
        choiceLabel.setBounds(50, 50, 100, 20);
        frame.add(choiceLabel);

        Choice choice = new Choice();
        choice.setBounds(160, 50, 150, 20);
        choice.add("Apple");
        choice.add("Banana");
        choice.add("Orange");
        frame.add(choice);

        // ----- Radio Buttons -----
        Label genderLabel = new Label("Select Gender:");
        genderLabel.setFont(new Font("Arial", Font.BOLD, 14));
        genderLabel.setBounds(50, 90, 120, 20);
        frame.add(genderLabel);

        CheckboxGroup genderGroup = new CheckboxGroup();

        Checkbox maleRadio = new Checkbox("Male", genderGroup, false);
        maleRadio.setFont(new Font("Arial", Font.BOLD, 14));
        maleRadio.setBounds(180, 90, 80, 20);

        Checkbox femaleRadio = new Checkbox("Female", genderGroup, false);
        femaleRadio.setFont(new Font("Arial", Font.BOLD, 14));
        femaleRadio.setBounds(280, 90, 100, 20);
        frame.add(maleRadio);
        frame.add(femaleRadio);
```

```

// ----- Checkbox -----
Checkbox checkbox = new Checkbox("I agree to the terms!");
checkbox.setFont(new Font("Arial", Font.BOLD, 13));
checkbox.setBounds(50, 130, 250, 30);
frame.add(checkbox);

// ----- Button to Show Selections -----
Button submitButton = new Button("Submit");
submitButton.setBounds(50, 170, 80, 30);
frame.add(submitButton);

// ----- Label to Display Results -----
Label resultLabel = new Label();
resultLabel.setBounds(50, 210, 400, 30);
frame.add(resultLabel);

// ----- Event Handling -----
submitButton.addActionListener(e -> {
    String selectedFruit = choice.getSelectedItem();
    String agreement = checkbox.getState() ?
        "Agreed": "Not Agreed";
    String gender = genderGroup.getSelectedCheckbox() != null ?
        genderGroup.getSelectedCheckbox().getLabel(): "Not Selected";

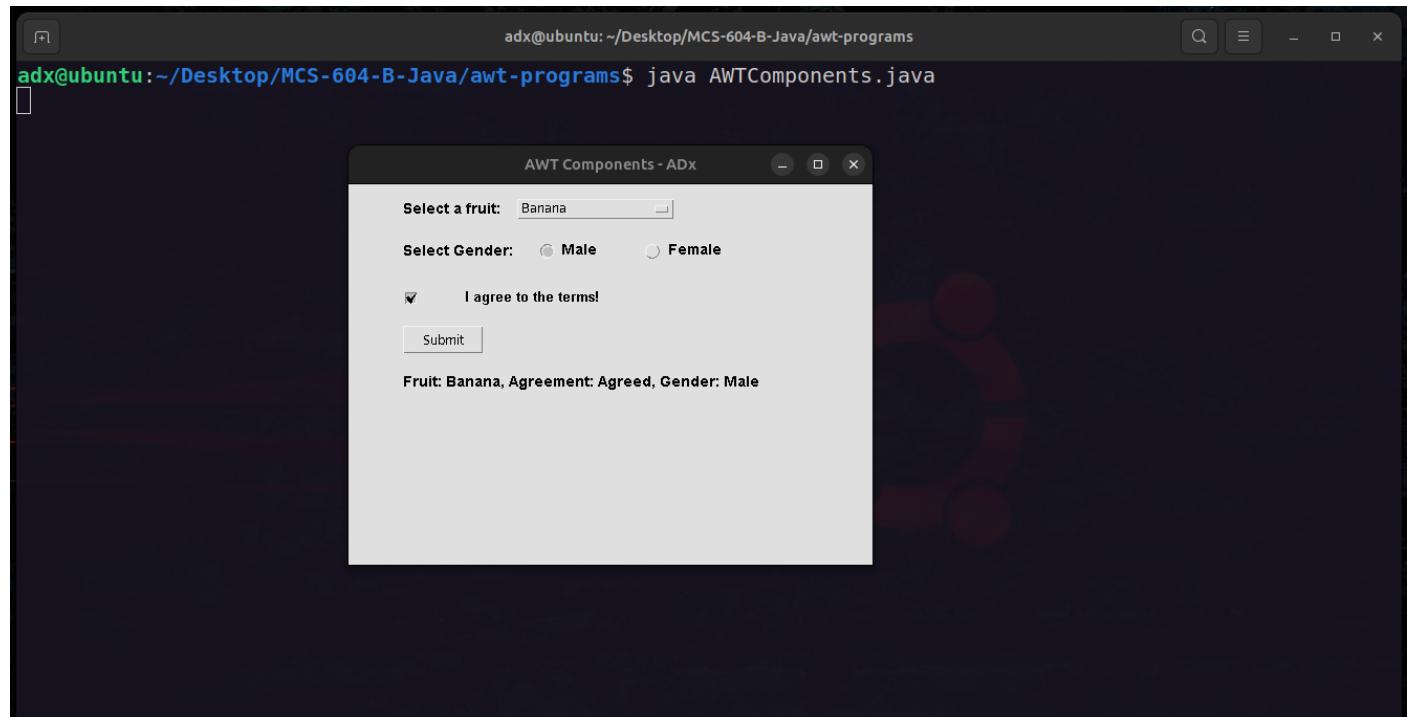
    resultLabel.setFont(new Font("Arial", Font.BOLD, 14));
    resultLabel.setText("Fruit: " + selectedFruit +
        ", Agreement: " + agreement + ", Gender: " + gender);
});

frame.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        frame.dispose();
        System.exit(0);
    }
});

frame.setVisible(true);
}
}

```

Output:



23. Write a program to implement LayoutManager. (AWT)

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

public class LayoutManager {
    public static void main(String[] args) {
        Frame frame = new Frame("LayoutManagers - ADx");
        frame.setSize(400, 300);

        // Create panel and buttons
        Panel panel = new Panel();
        Button btn1 = new Button("Button 1");
        Button btn2 = new Button("Button 2");
        Button btn3 = new Button("Button 3");
        Button btn4 = new Button("Button 4");
        Button btn5 = new Button("Button 5");

        // Add dropdown (Choice) to select LayoutManager
        Choice layoutChoice = new Choice();
        layoutChoice.add("FlowLayout");
        layoutChoice.add("BorderLayout");
        layoutChoice.add("GridLayout");

        // Event listener to switch layouts
        layoutChoice.addItemListener(e -> {
            panel.removeAll();
            panel.add(btn1);
            panel.add(btn2);
            panel.add(btn3);
            panel.add(btn4);
            panel.add(btn5);

            switch (layoutChoice.getSelectedItem()) {

                case "FlowLayout" ->
                    panel.setLayout(new FlowLayout());

                case "BorderLayout" -> {
                    panel.setLayout(new BorderLayout());
                    panel.add(btn1, BorderLayout.NORTH);
                    panel.add(btn2, BorderLayout.SOUTH);
                    panel.add(btn3, BorderLayout.EAST);
                    panel.add(btn4, BorderLayout.WEST);
                    panel.add(btn5, BorderLayout.CENTER);
                }
            }
        });
    }
}
```

```
        panel.add(btn2, BorderLayout.SOUTH);
        panel.add(btn3, BorderLayout.EAST);
        panel.add(btn4, BorderLayout.WEST);
        panel.add(btn5, BorderLayout.CENTER);
    }

    case "GridLayout" ->
        panel.setLayout(new GridLayout(2, 2));
    }
    panel.validate();
}) ;

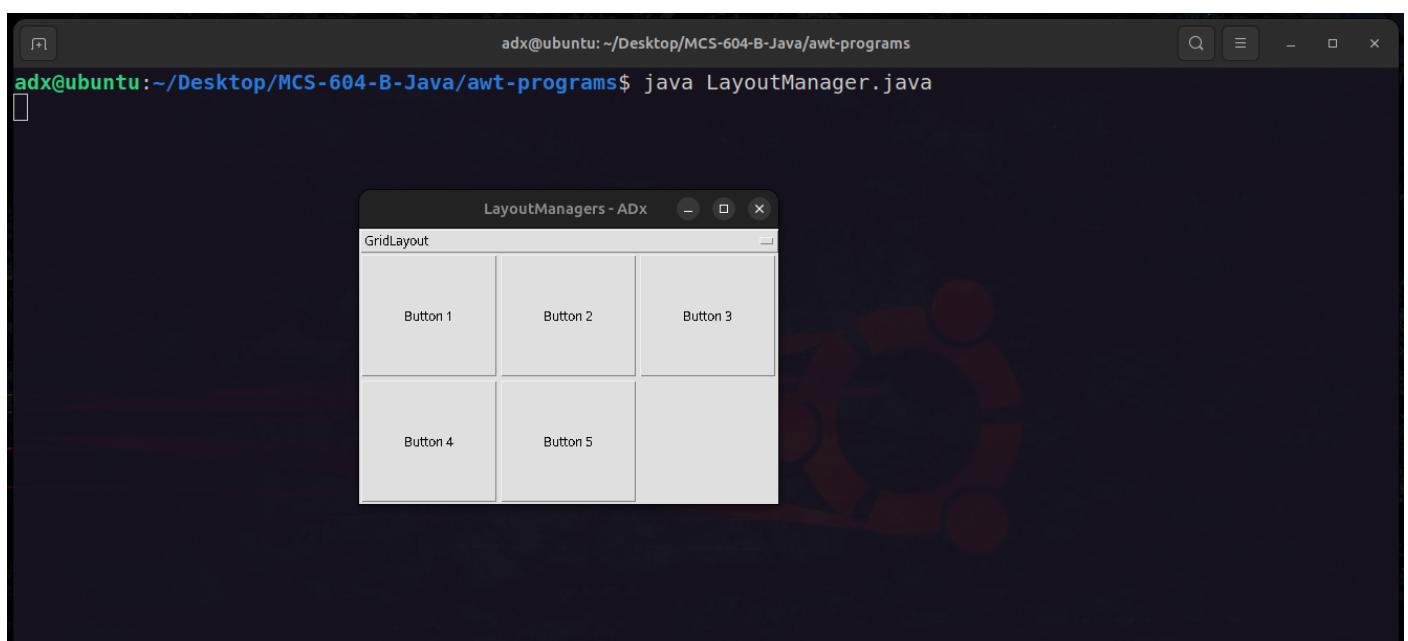
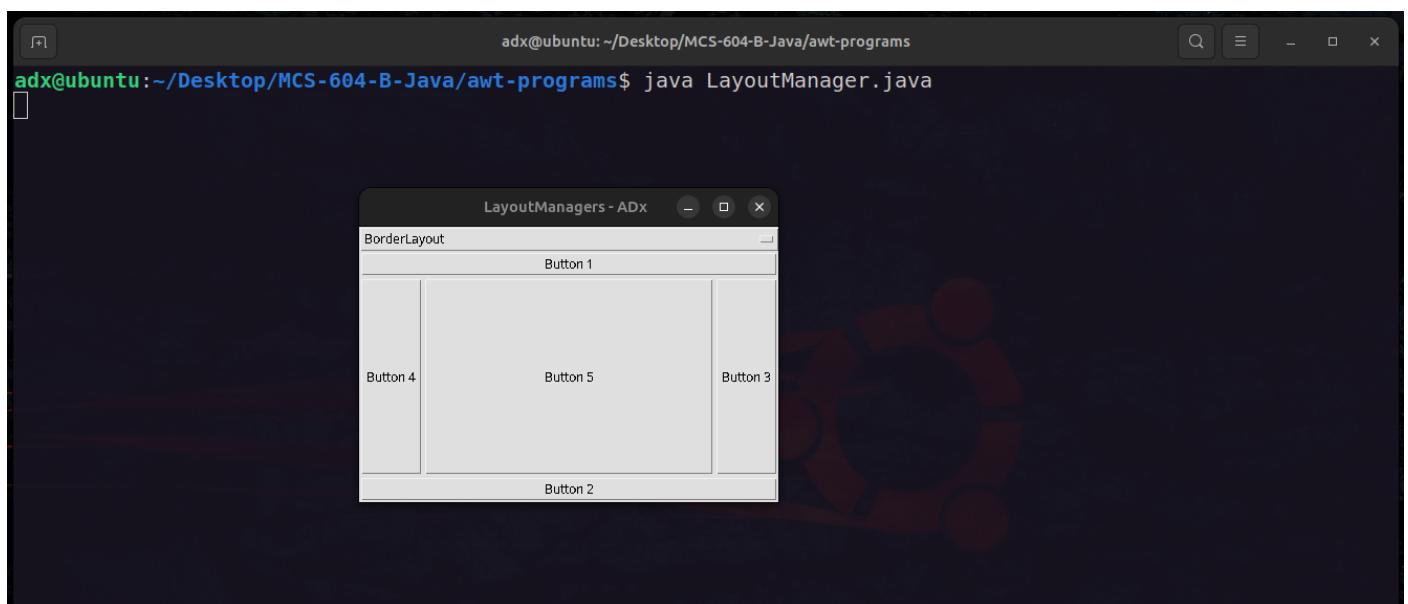
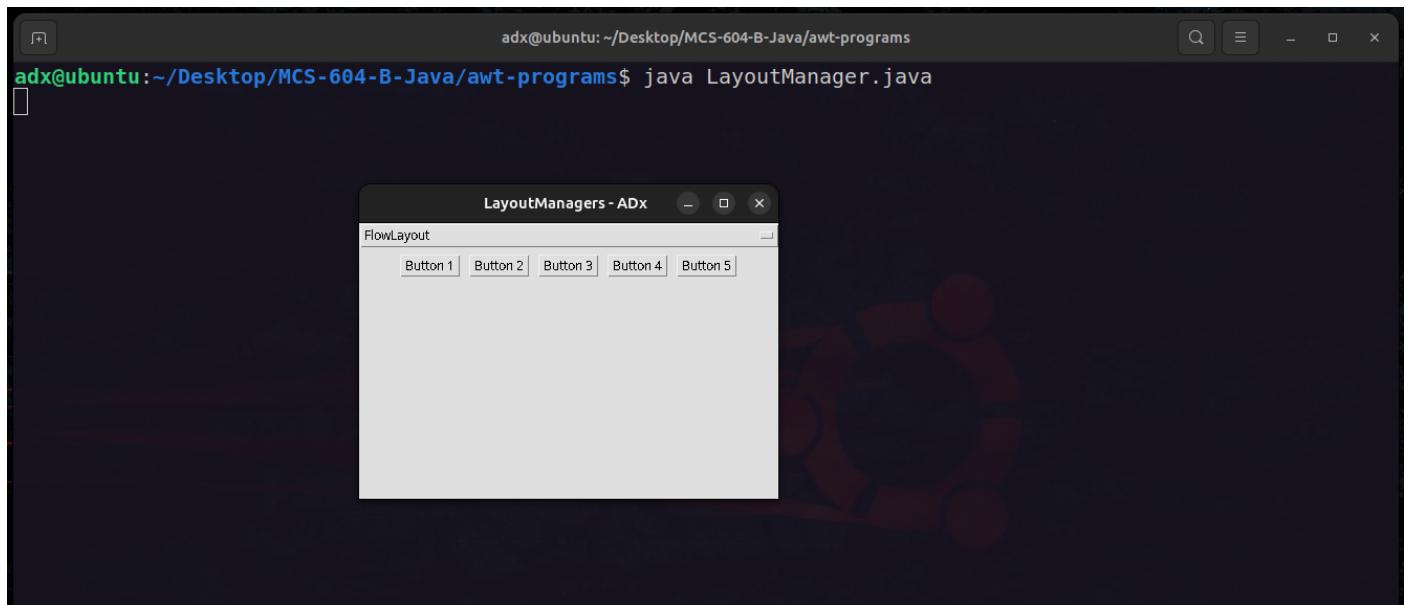
// Set default layout and add components
panel.setLayout(new FlowLayout());
panel.add(btn1);
panel.add(btn2);
panel.add(btn3);
panel.add(btn4);
panel.add(btn5);

// Add dropdown and panel to frame
frame.add(layoutChoice, BorderLayout.NORTH);
frame.add(panel, BorderLayout.CENTER);

// Handle window closing
frame.addWindowListener(new WindowAdapter() {
    public void windowClosing(WindowEvent e) {
        frame.dispose();
        System.exit(0);
    }
});

frame.setVisible(true);
}
}
```

Output:



24. Write a program to implement a Dialog box. (AWT)

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

public class DialogBox {
    public static void main(String[] args) {
        Frame frame = new Frame("Dialog - ADx");
        frame.setSize(400, 300);
        frame.setLayout(new GridBagLayout());
        Button button = new Button("Surprise me!");
        frame.add(button);

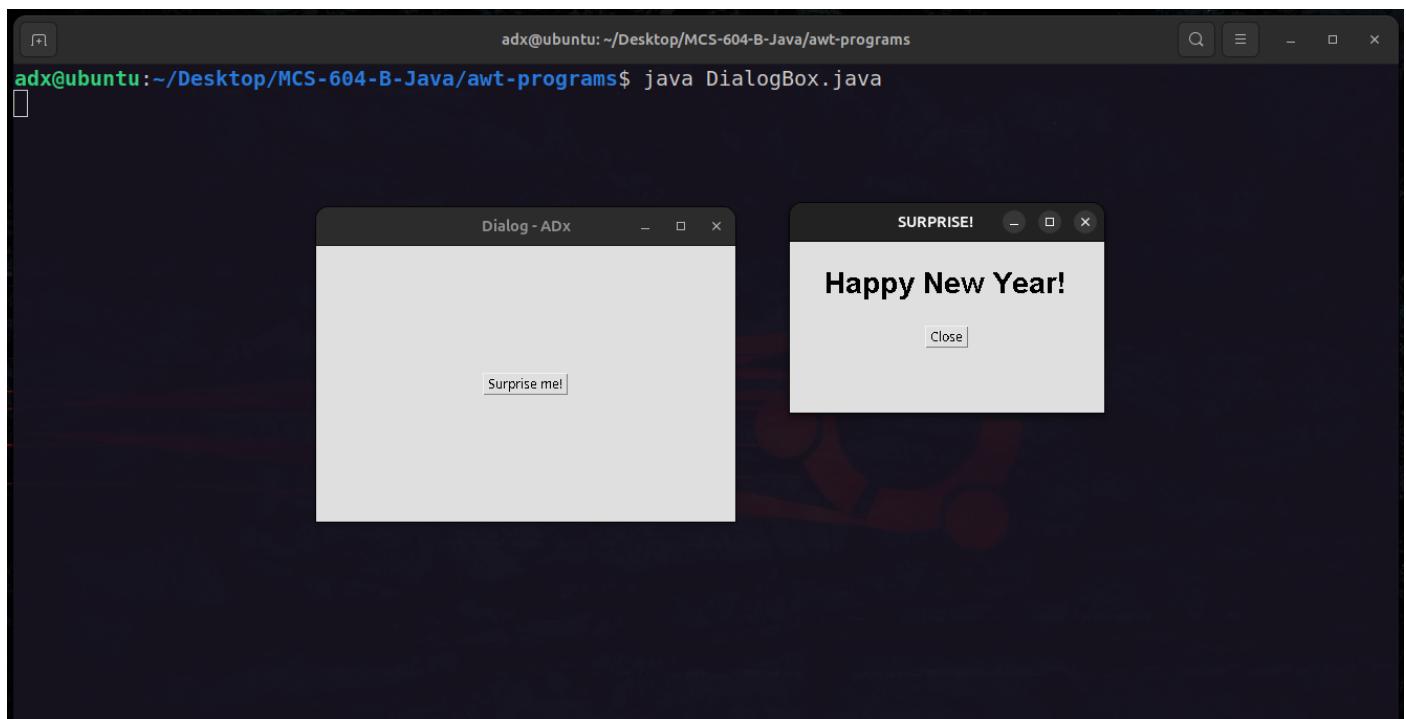
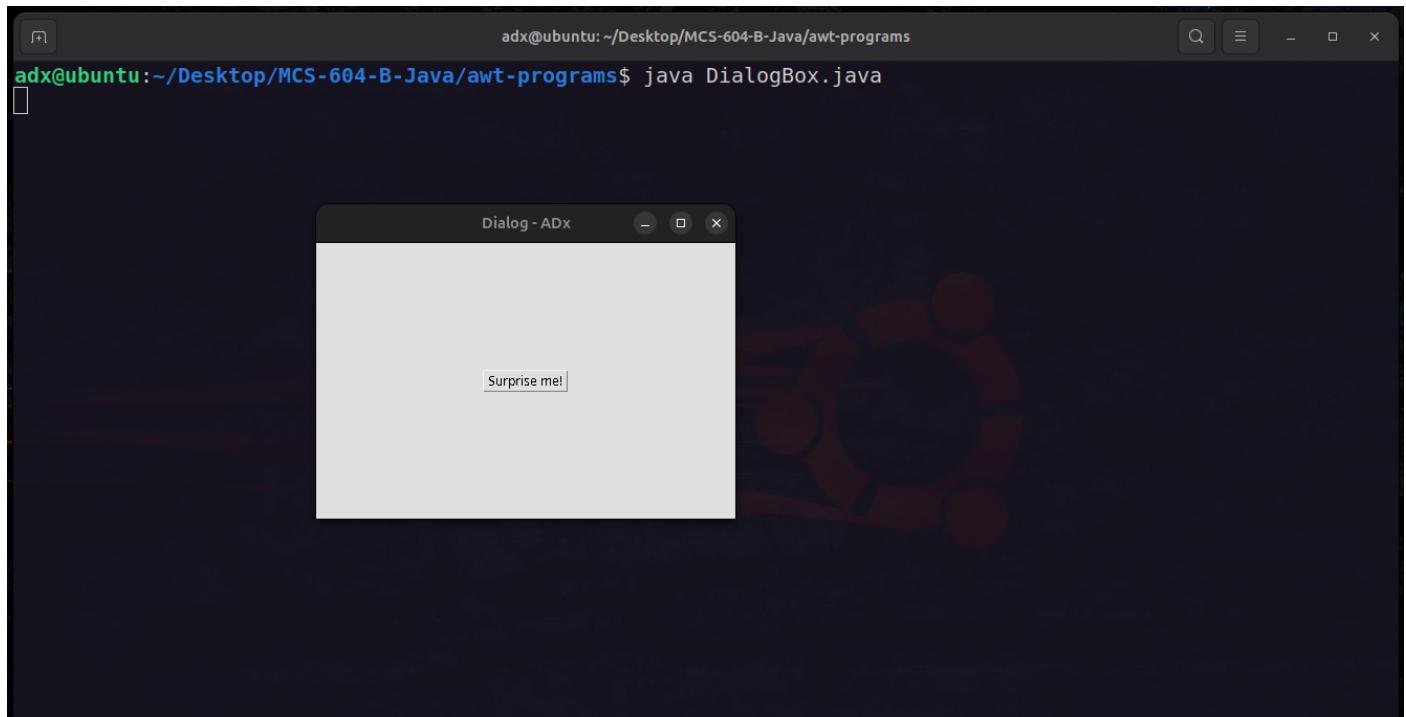
        // Create a modal dialog with a message and a close button
        Dialog dialog = new Dialog(frame, "SURPRISE!", true);
        dialog.setSize(300, 200);
        dialog.setLayout(new FlowLayout(FlowLayout.CENTER, 10, 20));

        // Add a label and a close button to the dialog
        Label message = new Label("Happy New Year!");
        message.setFont(new Font("Arial", Font.BOLD, 28));
        Button closeButton = new Button("Close");
        dialog.add(message);
        dialog.add(closeButton);

        // Event to show the dialog
        button.addActionListener(e -> dialog.setVisible(true));
        // Event to close the dialog
        closeButton.addActionListener(e -> dialog.setVisible(false));

        // Handle frame closing
        frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                frame.dispose();
                System.exit(0);
            }
        });
        frame.setVisible(true);
    }
}
```

Output:



25. Write a program to implement Smiley face. (AWT)

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

public class SmileyFace extends Frame {

    public SmileyFace() {
        setSize(400, 400);                      // Set the window size
        setTitle("Smiley Face-ADx"); // Set the window title
        setVisible(true);                      // Make the window visible
        setBackground(Color.BLACK); // Set the background color to black
    }

    // Override the paint method to draw the smiley face
    public void paint(Graphics g) {
        Graphics2D g2d = (Graphics2D) g;
        g2d.setColor(Color.YELLOW);
        g2d.fillOval(100, 100, 200, 200); // Draw face (circle)

        // Set color for the eyes (black)
        g2d.setColor(Color.BLACK);
        g2d.fillOval(150, 150, 30, 30); // Left eye
        g2d.fillOval(220, 150, 30, 30); // Right eye

        // Set color for the mouth (black)
        g2d.setColor(Color.BLACK);
        // Set a thicker stroke for the mouth
        g2d.setStroke(new BasicStroke(10));
        // Draw a smoother arc (bigger arc for a smoother curve)
        g2d.drawArc(130, 170, 140, 80, 0, -180); // Draw mouth (arc)
    }

    public static void main(String[] args) {
        SmileyFace smiley = new SmileyFace();
        // Handle window closing event
        smiley.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }
}
```

Output:

```
adx@ubuntu:~/Desktop/MCS-604-B-Java.awt-programs$ java SmileyFace.java
```



26. Write a program to implement System Clock. (AWT)

```
import java.awt.*;
import java.awt.event.*;
import java.text.SimpleDateFormat;
import java.util.Date;

public class SystemClock extends Frame {
    private Label timeLabel;

    public SystemClock() {
        setSize(400, 200);
        setTitle("System Clock - ADx");
        setLayout(new GridBagLayout());
        setVisible(true);

        // Create a label to display time
        timeLabel = new Label();
        timeLabel.setFont(new Font("Arial", Font.BOLD, 30));
        add(timeLabel);

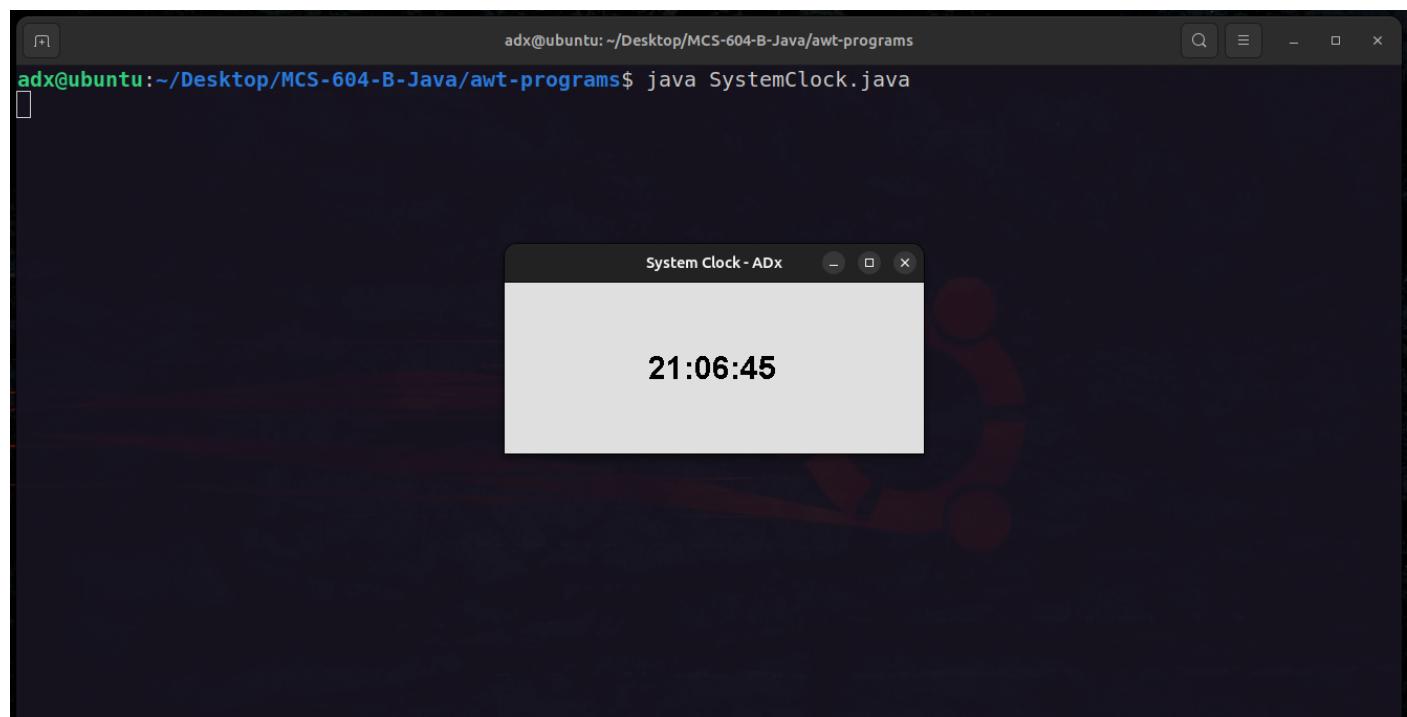
        // Handle window closing event
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        updateClock();
    }

    // Method to update the clock every second
    private void updateClock() {
        Thread clockThread = new Thread(() -> {
            while (true) {
                // Get the current system time
                SimpleDateFormat t = new SimpleDateFormat("HH:mm:ss");

                String time = t.format(new Date());
                timeLabel.setText(time);
            }
        });
    }
}
```

```
        try {
            // Wait for 1 second before updating the time again
            Thread.sleep(1000);
        }
        catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
});  
clockThread.start(); // Start the clock thread  
}  
  
public static void main(String[] args) {  
    new SystemClock();  
}  
}
```

Output:



27. Write a program to implement Inter-Thread Communication.

```
class SharedResource {
    private int data;
    private boolean isAvailable = false;

    // Method to produce data
    public synchronized void produce(int value) {
        while (isAvailable) {
            try {
                wait(); // Wait until the data is consumed
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        data = value;
        isAvailable = true;
        System.out.println("Produced: " + data);
        notify(); // Notify the consumer that data is available
    }

    // Method to consume data
    public synchronized void consume() {
        while (!isAvailable) {
            try {
                wait(); // Wait until the data is produced
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        System.out.println("Consumed: " + data);
        isAvailable = false;
        notify(); // Notify the producer that the data has been consumed
    }
}

class Producer extends Thread {
    private SharedResource sharedResource;
    public Producer(SharedResource sharedResource) {
        this.sharedResource = sharedResource;
    }
}
```

```

@Override
public void run() {
    for (int i = 1; i <= 5; i++) {
        sharedResource.produce(i);
        try {
            Thread.sleep(500);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

class Consumer extends Thread {
    private SharedResource sharedResource;
    public Consumer(SharedResource sharedResource) {
        this.sharedResource = sharedResource;
    }

    @Override
    public void run() {
        for (int i = 1; i <= 5; i++) {
            sharedResource.consume();
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class Solution27 {
    public static void main(String[] args) {
        SharedResource sharedResource = new SharedResource();
        Producer producer = new Producer(sharedResource);
        Consumer consumer = new Consumer(sharedResource);
        producer.start();
        consumer.start();
    }
}

```

Output:

```
adx@ubuntu:~/Desktop/MCS-604-B-Java$ java Solution27
Produced: 1
Consumed: 1
Produced: 2
Consumed: 2
Produced: 3
Consumed: 3
Produced: 4
Consumed: 4
Produced: 5
Consumed: 5
adx@ubuntu:~/Desktop/MCS-604-B-Java$
```

28. Write a program to create a Frame that displays the student information. (Swing)

```
import java.awt.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

public class Solution28 extends JFrame {

    public Solution28() {
        setTitle("Student Information");
        setSize(600, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        String[] header = {"Registration No", "Name", "Email", "Phone"};
        Object[][] data = {
            {"S101", "Alice Johnson", "alice@example.com", "9876543210"},
            {"S102", "Bob Smith", "bob@example.com", "9123456789"},
            {"S103", "Charlie Brown", "charl@example.com", "9988776655"},
            {"S104", "David White", "david@example.com", "9871234560"},
            {"S105", "Ella Green", "ella@example.com", "9234567890"}
        };

        DefaultTableModel tmodel = new DefaultTableModel(data, header);
        JTable studentTable = new JTable(tmodel);
        JScrollPane scrollPane = new JScrollPane(studentTable);
        add(scrollPane, BorderLayout.CENTER);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            Solution28 frame = new Solution28();
            frame.setVisible(true);
        });
    }
}
```

Output:

```
adx@ubuntu:~/Desktop/MCS-604-B-Java$ java Solution28
```

The screenshot shows a terminal window on a dark-themed desktop environment. The terminal title bar reads "adx@ubuntu: ~/Desktop/MCS-604-B-Java". Below the title bar, the command "java Solution28" is entered. A Java application window titled "Student Information" is displayed in the foreground. The window contains a table with five rows of student data, each consisting of four columns: Registration No, Name, Email, and Phone.

Registration No	Name	Email	Phone
S101	Alice Johnson	alice@example.com	9876543210
S102	Bob Smith	bob@example.com	9123456789
S103	Charlie Brown	charlie@example.com	9988776655
S104	David White	david@example.com	9871234560
S105	Ella Green	ella@example.com	9234567890

29. Write a simple JDBC program to retrieve student information after connecting to the database.

```
import java.sql.*;

class JDBC_CLI {
    private Connection con;
    // Database Credentials
    private static final String BASE = "jdbc:mysql://localhost:3306/";
    private static final String DB = "students_record";
    private static final String DB_URL = BASE + DB;
    private static final String DB_USER = "adx";
    private static final String DB_PASS = "adx";

    JDBC_CLI() {
        connect();
    }

    private void connect() {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(DB_URL, DB_USER, DB_PASS);
            System.out.println("Successfully connected to database!");
            viewStudents();
        }
        catch (ClassNotFoundException e) {
            System.out.println("MySQL JDBC Driver not found!");
        }
        catch (SQLException e) {
            System.out.println("Database connection failed!");
        }
    }

    private void printTable(ResultSet rs) throws SQLException {
        ResultSetMetaData metaData = rs.getMetaData();
        int columnCount = metaData.getColumnCount();
        System.out.println("TABLE:: " + metaData.getTableName(1) + "\n");
        for (int i = 1; i <= columnCount; i++) {
            if(i == 1) {
                System.out.printf("%-10s", metaData.getColumnName(i));
            }
        }
    }
}
```

```

        else {
            System.out.printf("%-30s", metaData.getColumnName(i));
        }
    }
System.out.println();
System.out.println("=".repeat(columnCount * 20));

// Print the data rows
while (rs.next()) {
    for (int i = 1; i <= columnCount; i++) {
        if(i == 1) {
            System.out.printf("%-10s", rs.getString(i));
        }
        else {
            System.out.printf("%-30s", rs.getString(i));
        }
    }
    System.out.println();
}
}

private void viewStudents() {
try {
    String query = "SELECT * FROM student_info";
    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery(query);
    printTable(rs);
}
catch (SQLException e) {
    System.out.println("Unable to retrieve details!");
}
}

public static void main(String[] args) {
    JDBC_CLI admin = new JDBC_CLI();
}
}

```

Output:

```
Successfully connected to database!
```

```
TABLE:: student_info
```

regn_no	name	email	phone
1	Anuj Das	anujdas@gmail.com	8638756810
2	Pragati Dey	deypragati@gmail.com	7577903238
3	Aditya Dutta	adi22@gmail.com	7006546789
4	Kaustav Paul	kosu@gmail.com	9394099856
5	Arkodyoti Choudhury	arko1766@gmail.com	8945659845
6	Bishwak Nath	bishwakn22@gmail.com	8984756254
7	Bips Pol	bips01@gmail.com	7965442652
8	Anne Marie	annemar@gmail.com	7002515462

```
Process finished with exit code 0
```

30. Write a program to create a GUI using Swing that performs database operations.

```
import java.sql.*;
import javax.swing.*;
import java.awt.event.*;
import java.text.MessageFormat;
import net.proteanit.sql.DbUtils;
import javax.swing.table.DefaultTableModel;

public class StudentsRecord extends JFrame implements ActionListener {
    // Database credentials
    private static final String BASE_URL = "jdbc:mysql://localhost:3306/";
    private static final String DB_NAME = "students_record";
    private static final String DB_URL = BASE_URL + DB_NAME;
    private static final String DB_USER = "adx";
    private static final String DB_PASSWORD = "adx";

    private Connection con;

    // Swing components
    private JTextField regn_no, name, email, phone, search_by_regn_no;
    private JButton create, printb, update, delete, search, reset, exit;
    private JTable tab_data = new JTable();
    private JScrollPane table_Panel;
    // Constructor
    public StudentsRecord() {
        super("Student Records Dashboard");
        connectToDatabase();
        initializeGUI();
        loadTable("SELECT * FROM student_info");
    }

    private void loadTable(String query) {
        try {
            PreparedStatement pst = con.prepareStatement(query);
            ResultSet rs = pst.executeQuery();
            tab_data.setModel(DbUtils.resultSetToTableModel(rs));
            table_Panel.setViewportView(tab_data);
        }
        catch (SQLException e) {
            JOptionPane.showMessageDialog(null,
                "Error loading data: " + e.getMessage(),
                "Database Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

```

private void connectToDatabase() {
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        con = DriverManager.getConnection(
            DB_URL,
            DB_USER,
            DB_PASSWORD
        );
        System.out.println("Connection Successful!");
    }
    catch (ClassNotFoundException e) {
        JOptionPane.showMessageDialog(this,
            "MySQL JDBC Driver not found: " + e.getMessage(),
            "Driver Error", JOptionPane.ERROR_MESSAGE);
    }
    catch (SQLException e) {
        JOptionPane.showMessageDialog(this,
            "Database con failed: " + e.getMessage(),
            "Connection Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void initializeGUI() {
    // Set frame properties
    setLayout(null);
    setVisible(true);
    setLocationRelativeTo(null);
    setSize(800, 650);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    // Create labels and text fields
    JLabel regn_no_label = new JLabel("Registration Number:");
    regn_no_label.setBounds(50, 50, 150, 35);
    add(regn_no_label);

    regn_no = new JTextField();
    regn_no.setBounds(200, 50, 300, 35);
    add(regn_no);

    JLabel name_label = new JLabel("Full Name:");
    name_label.setBounds(50, 100, 150, 35);
    add(name_label);

    name = new JTextField();
    name.setBounds(200, 100, 300, 35);
    add(name);

    JLabel email_label = new JLabel("Email:");
    email_label.setBounds(50, 150, 150, 35);
    add(email_label);
}

```

```
email = new JTextField();
email.setBounds(200, 150, 300, 35);
add(email);

JLabel phone_label = new JLabel("Phone Number:");
phone_label.setBounds(50, 200, 150, 35);
add(phone_label);

phone = new JTextField();
phone.setBounds(200, 200, 300, 35);
add(phone);

JLabel search_label = new JLabel("Search by Registration Number:");
search_label.setBounds(50, 300, 250, 35);
add(search_label);

search_by_regn_no = new JTextField();
search_by_regn_no.setBounds(250, 300, 250, 35);
add(search_by_regn_no);

// Create and position buttons
create = new JButton("Add New");
create.setBounds(550, 50, 100, 35);
create.addActionListener(this);
add(create);

search = new JButton("Search");
search.setBounds(550, 300, 100, 35);
search.addActionListener(this);
add(search);

update = new JButton("Update");
update.setBounds(550, 100, 100, 35);
update.addActionListener(this);
add(update);

delete = new JButton("Delete");
delete.setBounds(550, 150, 100, 35);
delete.addActionListener(this);
add(delete);

reset = new JButton("Reset");
reset.setBounds(550, 200, 100, 35);
reset.addActionListener(this);
add(reset);

printb = new JButton("Print");
printb.setBounds(550, 250, 100, 35);
printb.addActionListener(new ActionListener() {
```

```

@Override
public void actionPerformed(ActionEvent e) {
    String hmsg = "Printing in Progress";
    String fmsg = "Page-{0, number, integer}";
    MessageFormat headr = new MessageFormat(hmsg);
    MessageFormat footr = new MessageFormat(fmsg);
    try {
        tab_data.print(JTable.PrintMode.NORMAL, headr, footr);
    }
    catch (Exception ex) {
        throw new RuntimeException(ex);
    }
}
});

add(printb);

exit = new JButton("Exit");
exit.setBounds(670, 300, 80, 35);
exit.addActionListener(this);
add(exit);

// Initialize and set bounds for table and scroll pane
table_Panel = new JScrollPane(tab_data);
table_Panel.setBounds(50, 380, 700, 200);
tab_data.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        super.mouseClicked(e);
        DefaultTableModel rec = (DefaultTableModel) tab_data.getModel();
        int selectedRow = tab_data.getSelectedRow();
        String s_regn, s_name, s_email, s_phone;
        s_regn = rec.getValueAt(selectedRow, 0).toString();
        s_name = rec.getValueAt(selectedRow, 1).toString();
        s_email = rec.getValueAt(selectedRow, 2).toString();
        s_phone = rec.getValueAt(selectedRow, 3).toString();
        regn_no.setText(s_regn);
        name.setText(s_name);
        email.setText(s_email);
        phone.setText(s_phone);
    }
});
add(table_Panel);
}

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == exit) {
        try {
            if (con != null && !con.isClosed()) {
                con.close();
            }
        }
    }
}

```

```

        catch (SQLException ex) {
            ex.printStackTrace();
        }
        System.exit(0);
    }

    else if (e.getSource() == create) {
        addStudent();
    }

    else if (e.getSource() == update) {
        updateStudent();
    }

    else if (e.getSource() == delete) {
        deleteStudent();
    }

    else if (e.getSource() == reset) {
        resetFields();
    }

    else if(e.getSource() == search) {
        searchStudent();
    }
}

private void resetFields() {
    regn_no.setText("");
    name.setText("");
    email.setText("");
    phone.setText("");
    search_by_regn_no.setText("");
    loadTable("SELECT * FROM student_info");
}

private void addStudent() {
    try {
        String insertQuery =
        "INSERT INTO student_info (regn_no, name, email, phone) VALUES (?,?,?,?,?)";
        PreparedStatement pstmt = con.prepareStatement(insertQuery);
        pstmt.setString(1, regn_no.getText());
        pstmt.setString(2, name.getText());
        pstmt.setString(3, email.getText());
        pstmt.setString(4, phone.getText());
        int rowsAffected = pstmt.executeUpdate();
        JOptionPane.showMessageDialog(this,
            rowsAffected + " student record added successfully!");

        loadTable("SELECT * FROM student_info");
        resetFields();
    }

    catch (SQLException ex) {
        JOptionPane.showMessageDialog(this,
            "Error adding student: " + ex.getMessage(),
            "Database Error", JOptionPane.ERROR_MESSAGE);
    }
}

```

```

        }

    }

private void updateStudent() {
    try {
        String updateQuery =
        "UPDATE student_info SET name = ?, email = ?, phone = ? WHERE regn_no = ?";
        PreparedStatement pstmt = con.prepareStatement(updateQuery);
        pstmt.setString(1, name.getText());
        pstmt.setString(2, email.getText());
        pstmt.setString(3, phone.getText());
        pstmt.setString(4, regn_no.getText());

        int rowsAffected = pstmt.executeUpdate();
        JOptionPane.showMessageDialog(this,
            rowsAffected + " student record updated successfully!");

        loadTable("SELECT * FROM student_info");
        resetFields();
    }
    catch (SQLException ex) {
        JOptionPane.showMessageDialog(this,
            "Error updating student: " + ex.getMessage(),
            "Database Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void deleteStudent() {
    try {
        String deleteQuery = "DELETE FROM student_info WHERE regn_no = ?";
        PreparedStatement pstmt = con.prepareStatement(deleteQuery);
        pstmt.setString(1, regn_no.getText());

        int rowsAffected = pstmt.executeUpdate();
        JOptionPane.showMessageDialog(this,
            rowsAffected + " student record deleted successfully!");

        loadTable("SELECT * FROM student_info");
        resetFields();
    }
    catch (SQLException ex) {
        JOptionPane.showMessageDialog(this,
            "Error deleting student: " + ex.getMessage(),
            "Database Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void searchStudent() {
    try {
        String searchQuery = "SELECT * FROM student_info WHERE regn_no = ?";
        PreparedStatement pstmt = con.prepareStatement(searchQuery);

```

```
pstmt.setString(1, search_by_regn_no.getText()) ;
ResultSet rs = pstmt.executeQuery();

if (rs.next()) {
    String stud_regn = rs.getString("regn_no");
    loadTable("SELECT * FROM student_info WHERE regn_no = " + stud_regn);
    regn_no.setText(stud_regn);
    name.setText(rs.getString("name"));
    email.setText(rs.getString("email"));
    phone.setText(rs.getString("phone"));
    JOptionPane.showMessageDialog(this, "Record found!");
}
else {
    JOptionPane.showMessageDialog(this,
        "No record found with the given Registration Number.");
}
}

catch (SQLException ex) {
    JOptionPane.showMessageDialog(this,
        "Error searching student: " + ex.getMessage(),
        "Database Error", JOptionPane.ERROR_MESSAGE);
}
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new StudentsRecord());
}
}
```

Output:

Student Records Dashboard

Registration Number:	<input type="text"/>	Add New
Full Name:	<input type="text"/>	Update
Email:	<input type="text"/>	Delete
Phone Number:	<input type="text"/>	Reset
Search by Registration Number: <input type="text"/>		Print
		Search
		Exit

regn_no	name	email	phone
1	Anuj Das	anujdas@gmail.com	8638756810
2	Pragati Dey	deypragati@gmail.com	7577903238
3	Aditya Dutta	adi22@gmail.com	7006546789
4	Kaustav Paul	kosu@gmail.com	9394099856
5	Arkodyoti Choudhury	arko1766@gmail.com	8945659845
6	Bishwak Nath	bishwakn22@gmail.com	8984756254
7	Biprojyoti Paul	bips01@gmail.com	7965442652
8	Anne Marie	annemar@gmail.com	7002515462
10	Mike Shinoda	mike.lp@gmail.com	6546865231
11	Reeya Dey	reeya.yeo.boo@gmail.com	7055694235
12	Nina William	nina.tekken@gmail.com	6543652635

Student Records Dashboard

Registration Number:	<input type="text"/> 1	Add New
Full Name:	<input type="text"/> Anuj Das	Update
Email:	<input type="text"/> anujdas2000@gmail.com	Delete
Phone Number:	<input type="text"/> 8638756810	Reset
Search by Registration Number: <input type="text"/>		Print
		Search
		Exit

Message

i 1 student record updated successfully!

OK

regn_no	name	email	phone
1	Anuj Das	anujdas@gmail.com	8638756810
2	Pragati Dey	deypragati@gmail.com	7577903238
3	Aditya Dutta	adi22@gmail.com	7006546789
4	Kaustav Paul	kosu@gmail.com	9394099856
5	Arkodyoti Choudhury	arko1766@gmail.com	8945659845
6	Bishwak Nath	bishwakn22@gmail.com	8984756254
7	Biprojyoti Paul	bips01@gmail.com	7965442652
8	Anne Marie	annemar@gmail.com	7002515462
10	Mike Shinoda	mike.lp@gmail.com	6546865231
11	Reeya Dey	reeya.yeo.boo@gmail.com	7055694235
12	Nina William	nina.tekken@gmail.com	6543652635