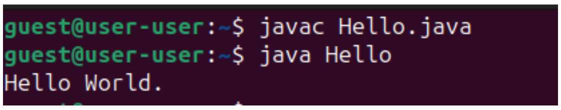
Week-I

1. Write a Java program print “Hello World”

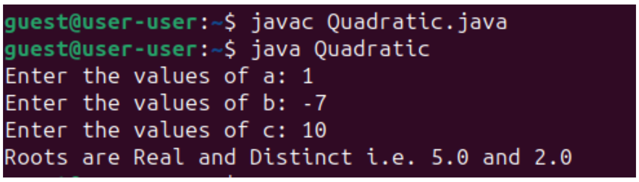
```
public class Hello{
public static void main(String[] args)
{
   System.out.println("Hello World.");
}
}
```



```
guest@user-user:~$ javac Hello.java
guest@user-user:~$ java Hello
Hello World.
```

2. Write a Java program that prints all real and imaginary solutions to the quadratic equation ax2 + bx + c = 0. Read in a, b, c and use the quadratic formula

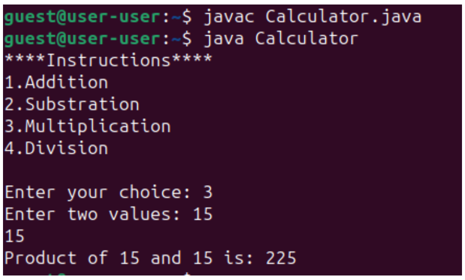
```
import java.util.*;
public class Quadratic{
  public static void main (String[] args) {
    int a,b,c;
    Scanner scan = new Scanner(System.in);
    Algebra quadratic = new Algebra();
    System.out.print("Enter the values of a: ");
    a = scan.nextInt();
    System.out.print("Enter the values of b: ");
    b = scan.nextInt();
    System.out.print("Enter the values of c: ");
    c = scan.nextInt();
    quadratic.findRoots(a,b,c);
    scan.close();
  }
}

class Algebra{
  void findRoots(int a, int b, int c) {
    double descr, root1, root2;
    descr = (b*b) - (4*a*c);
    root1 = (double)-b/(2*a);
    if(descr == 0)
      System.out.println("Roots are Real and Equal i.e. "+root1);
    else if(descr > 0){
      root2 = (double) Math.sqrt(descr)/(2*a);
      System.out.println("Roots are Real and Distinct i.e. "+(root1+root2)+" and "+(root1-root2));
    }
    else {
      root2 = (double) Math.sqrt(-descr)/(2*a);
      System.out.println("Roots are Imaginary i.e. "+root1+"+"+root2+"i and "+root1+"-"+root2+"i");
    }
  }
}
```



```
guest@user-user:~$ javac Quadratic.java
guest@user-user:~$ java Quadratic
Enter the values of a: 1
Enter the values of b: -7
Enter the values of c: 10
Roots are Real and Distinct i.e. 5.0 and 2.0
```

3. Write a Java program to implement calculator operations

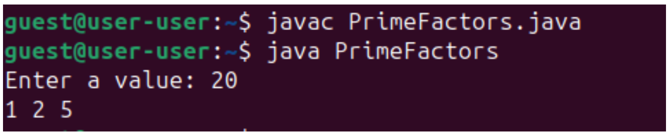
```
import java.util.*;

public class Calculator {
  public static void main (String[] args) {
    int choice, value1, value2;
    Scanner scan = new Scanner (System.in);
    Opertations opcode = new Opertations();
          System.out.print("****Instructions****\n1.Addition\n2.Substration\n3.Multiplication\n4.Division\n\nEnter your choice: ");
    choice = scan.nextInt();
    System.out.print("Enter two values: ");
    value1 = scan.nextInt();
    value2 = scan.nextInt();
    switch(choice) {
      case 1 -> opcode.add(value1,value2);
      case 2 -> opcode.substract(value1,value2);
      case 3 -> opcode.product(value1,value2);
      case 4 -> opcode.quotient(value1,value2);
      default -> System.out.println("Invalid choice");
    }
  }
}

class Opertations {
  void add(int num1, int num2) {
    System.out.println("Sum of "+num1+" and "+num2+" is: "+(num1+num2));
  }
  void substract(int num1, int num2) {
    System.out.println(num1+" - "+num2+" is: "+(num1-num2));
  }
  void product(int num1, int num2) {
    System.out.println("Product of "+num1+" and "+num2+" is: "+(num1*num2));
  }
  void quotient(int num1, int num2) {
    if(num2 == 0 || (num1 == 0 && num2 == 0))
      System.out.println("Division by zero error");
    else
      System.out.println(num1+"/"+num2+" is: "+(num1/num2));
  }
}
```



```
guest@user-user:~$ javac Calculator.java
guest@user-user:~$ java Calculator
****Instructions****
1.Addition
2.Substration
3.Multiplication
4.Division

Enter your choice: 3
Enter two values: 15
15
Product of 15 and 15 is: 225
```

4. Write a java program to find prime factors of given number

```
import java.util.*;

public class PrimeFactors {
  public static void main (String[] args) {
    int value;
    Scanner scan = new Scanner(System.in);
    Mathematics math = new Mathematics();
    System.out.print("Enter a value: ");
    value = scan.nextInt();
    math.findPrimeFactors(value);
    scan.close();
  }
}
```

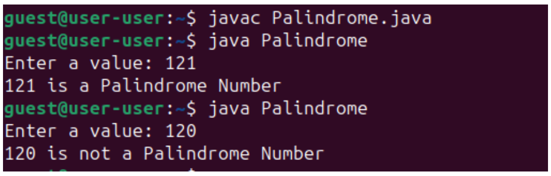


```
guest@user-user:~$ javac PrimeFactors.java
guest@user-user:~$ java PrimeFactors
Enter a value: 20
1 2 5
```

```
class Mathematics {
  void findPrimeFactors(int num) {
    int result;
    if (num < 0)
      System.out.println("Prime factors exists only for Positive Numbers");
    else {
      for(int i = 1; i <= num && num != 0 ; i++) {
        if(num%i == 0) {
          result = checkPrime(i);
          if(result != -1)
            System.out.print(result+" ");
          num /= i;
        }

      }
    }
  }

  int checkPrime(int value) {
    for (int i = 2; i < Math.sqrt(value) ; i++ ) {
      if((value % i) == 0 )
        return -1;
    }
    return value;
  }
}
```

5. Write a java program to find whether given number is Palindrome or not

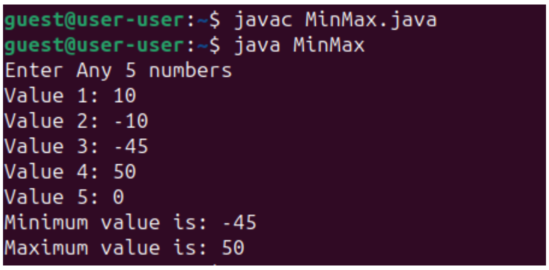
```
import java.util.*;

public class Palindrome {
  public static void main (String[] args) {
    int value;
    Scanner scan = new Scanner(System.in);
    Logic log = new Logic();
    System.out.print("Enter a value: ");
    value = scan.nextInt();
    log.checkPalindrome(value);
    scan.close();
  }
}

class Logic{
  void checkPalindrome(int num) {
    int result = num, sum = 0;
    while(num > 0) {
      sum = sum*10 + (num%10);
      num /= 10;
    }
    if(result == sum)
      System.out.println(result+" is a Palindrome Number");
    else
      System.out.println(result+" is not a Palindrome Number");
  }
}
```



```
guest@user-user:~$ javac Palindrome.java
guest@user-user:~$ java Palindrome
Enter a value: 121
121 is a Palindrome Number
guest@user-user:~$ java Palindrome
Enter a value: 120
120 is not a Palindrome Number
```

6. Write an application that declares 5 integers, determines and prints the largest and smallest in the group.

```
import java.util.*;

public class MinMax {
  public static void main (String[] args) {
    Scanner scan = new Scanner(System.in);
    Calculation cal = new Calculation();
    int []values = new int[5];
    System.out.println("Enter Any 5 numbers ");
    for(int i = 0; i < 5; i++){
      System.out.print("Value "+(i+1)+": ");
      values[i] = scan.nextInt();
    }
    cal.findMaxAndMinValues(values);
    scan.close();
  }
}

class Calculation{
  void findMaxAndMinValues(int []nums) {
    int MIN = nums[0], MAX = nums [0];
    for(int i = 1; i < 5; i++) {
      if(MIN > nums[i])
        MIN = nums[i];
      if(MAX < nums[i])
        MAX = nums[i];
    }
    System.out.println("Minimum value is: "+MIN+"\nMaximum value is: "+MAX);
  }
}
```

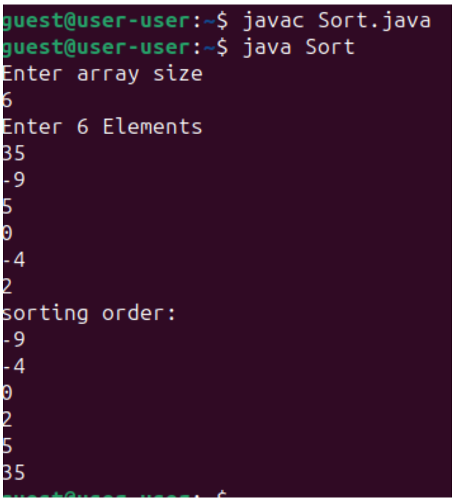


```
guest@user-user:~$ javac MinMax.java
guest@user-user:~$ java MinMax
Enter Any 5 numbers
Value 1: 10
Value 2: -10
Value 3: -45
Value 4: 50
Value 5: 0
Minimum value is: -45
Maximum value is: 50
```

Week-II

1. Write a Java program to sort a given list of numbers.

```
import java.util.Scanner;

class Sort {
    public static void main(String arg[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter array size");
        int n = sc.nextInt();
        int a[] = new int[n];
        System.out.println("Enter " + n + " Elements");
        for (int i = 0; i < n; i++) {
            a[i] = sc.nextInt();
        }
        for (int i = 0; i < n; i++) {
            for (int j = i + 1; j < n; j++) {
                if (a[i] > a[j]) {
                    int temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
```

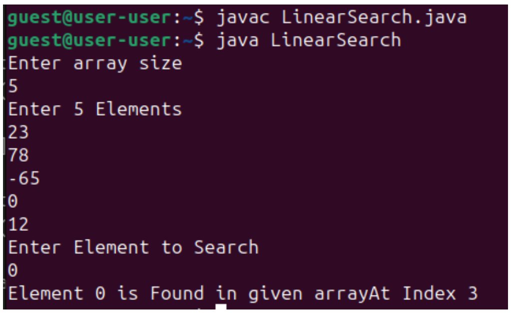


```
guest@user-user:~$ javac Sort.java
guest@user-user:~$ java Sort
Enter array size
6
Enter 6 Elements
35
-9
5
0
-4
2
sorting order:
-9
-4
0
2
5
35
```

```
                }
            }
        }
        System.out.println("sorting order:");
        for (int i = 0; i < n - 1; i++) {
            System.out.println(a[i] + " ");
        }
        System.out.println(a[n - 1]);
    }
}
```

2. Write a Java program to implement linear search.

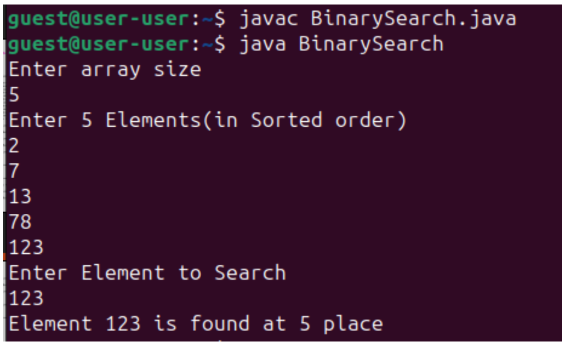
```
import java.util.Scanner;

class LinearSearch {
    public static void main(String arg[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter array size");
        int n = sc.nextInt();
        int a[] = new int[n];
        int i;

        System.out.println("Enter " + n + " Elements");
        for (i = 0; i < n; i++) {
            a[i] = sc.nextInt();
        }

        System.out.println("Enter Element to Search");
        int search = sc.nextInt();

        boolean found = false;
        for (i = 0; i < n; i++) {
            if (a[i] == search) {
                System.out.println("Element " + search + " is Found in the given array at Index " + i);
                found = true;
                break;
            }
        }

        if (!found) {
            System.out.println("Element " + search + " is Not Found in the given array");
        }
    }
}
```



```
guest@user-user:~$ javac LinearSearch.java
guest@user-user:~$ java LinearSearch
Enter array size
5
Enter 5 Elements
23
78
-65
0
12
Enter Element to Search
0
Element 0 is Found in given arrayAt Index 3
```

3. Write a Java program to implement binary search.

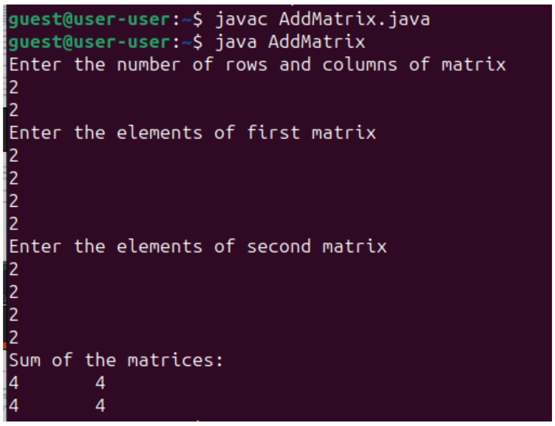
```
import java.util.Scanner;

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

        System.out.println("Enter array size:");
        int n = sc.nextInt();
        int a[] = new int[n];
        System.out.println("Enter " + n + " Elements (in sorted order):");
        for (int i = 0; i < n; i++) {
            a[i] = sc.nextInt();
        }

        System.out.println("Enter Element to Search:");
        int search = sc.nextInt();

        int first = 0, last = n - 1, middle;

        while (first <= last) {
            middle = (first + last) / 2;

            if (a[middle] < search) {
                first = middle + 1; // Move to the right half
            } else if (a[middle] == search) {
                System.out.println("Element " + search + " is found at index " + middle);
                return; // Exit after finding the element
            } else {
                last = middle - 1; // Move to the left half
            }
        }

        System.out.println("Element " + search + " is not present in the array");
    }
}
```



```
guest@user-user:~$ javac BinarySearch.java
guest@user-user:~$ java BinarySearch
Enter array size
5
Enter 5 Elements(in Sorted order)
2
7
13
78
123
Enter Element to Search
123
Element 123 is found at 5 place
```

4. Write a java program to add two given matrices.

```
import java.util.Scanner;

class AddMatrix {
    public static void main(String args[]) {
        int m, n, c, d;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the number of rows:");
        m = in.nextInt();
        System.out.println("Enter the number of columns:");
        n = in.nextInt();

        int first[][] = new int[m][n];
        int second[][] = new int[m][n];
        int sum[][] = new int[m][n];

        System.out.println("Enter the elements of the first matrix:");
        for (c = 0; c < m; c++)
            for (d = 0; d < n; d++)
                first[c][d] = in.nextInt();

        System.out.println("Enter the elements of the second matrix:");
        for (c = 0; c < m; c++)
            for (d = 0; d < n; d++)
                second[c][d] = in.nextInt();

        for (c = 0; c < m; c++)
            for (d = 0; d < n; d++)
```

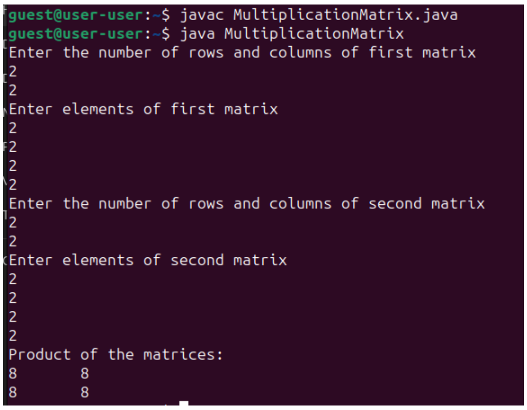


```
guest@user-user:~$ javac AddMatrix.java
guest@user-user:~$ java AddMatrix
Enter the number of rows and columns of matrix
2
2
Enter the elements of first matrix
2
2
2
2
Enter the elements of second matrix
2
2
2
2
Sum of the matrices:
4       4
4       4
```

```
                sum[c][d] = first[c][d] + second[c][d];

        System.out.println("Sum of the matrices:");
        for (c = 0; c < m; c++) {
            for (d = 0; d < n; d++)
                System.out.print(sum[c][d] + "\t");
            System.out.println();
        }
    }
}
```

5. Write a java program to multiply two given matrices.

```
import java.util.Scanner;

class MultiplicationMatrix {
    public static void main(String args[]) {
        int m, n, p, q, sum = 0, c, d, k;
        Scanner in = new Scanner(System.in);

        System.out.println("Enter the number of rows and columns of the first matrix:");
        m = in.nextInt();
        n = in.nextInt();
        int first[][] = new int[m][n];

        System.out.println("Enter elements of the first matrix:");
        for (c = 0; c < m; c++) {
            for (d = 0; d < n; d++) {
                first[c][d] = in.nextInt();
            }
        }

        System.out.println("Enter the number of rows and columns of the second matrix:");
        p = in.nextInt();
        q = in.nextInt();

        if (n != p) {
            System.out.println("The matrices can't be multiplied with each other.");
        } else {
            int second[][] = new int[p][q];
            int multiply[][] = new int[m][q];

            System.out.println("Enter elements of the second matrix:");
            for (c = 0; c < p; c++) {
                for (d = 0; d < q; d++) {
                    second[c][d] = in.nextInt();
                }
            }

            for (c = 0; c < m; c++) {
                for (d = 0; d < q; d++) {
                    for (k = 0; k < n; k++) { // Note: Corrected condition to `k < n`
                        sum += first[c][k] * second[k][d];
                    }
                    multiply[c][d] = sum;
                    sum = 0;
                }
```

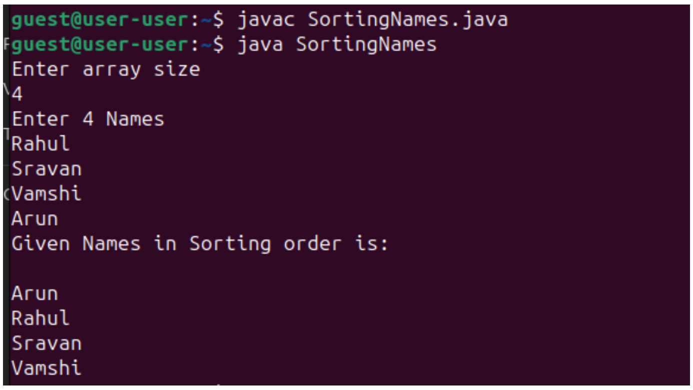


```
guest@user-user:~$ javac MultiplicationMatrix.java
guest@user-user:~$ java MultiplicationMatrix
Enter the number of rows and columns of first matrix
2
2
Enter elements of first matrix
2
2
2
2
Enter the number of rows and columns of second matrix
2
2
Enter elements of second matrix
2
2
2
2
Product of the matrices:
8       8
8       8
```

```
}

     System.out.println("Product of the matrices:");
     for (c = 0; c < m; c++) {
       for (d = 0; d < q; d++) {
         System.out.print(multiply[c][d] + "\t");
       }
       System.out.println();
     }
   }
 }
}
```

6. Write a java program for sorting a given list of names.

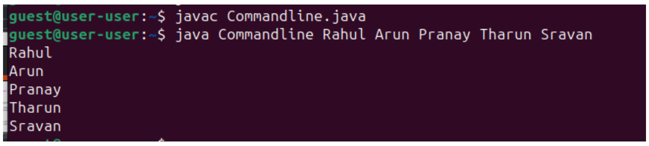
```
import java.util.Scanner;

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

        System.out.println("Enter array size:");
        int s = sc.nextInt();
        sc.nextLine(); // Consume the newline character left by nextInt()

        String[] a = new String[s];
        System.out.println("Enter " + s + " Names:");

        for (int i = 0; i < s; i++) {
            a[i] = sc.nextLine();
        }

        // Sorting the names in lexicographical order
        for (int i = 0; i < s; i++) {
            for (int j = i + 1; j < s; j++) {
                if (a[i].compareTo(a[j]) > 0) {
                    String temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }

        System.out.println("Given Names in Sorting Order:");
        for (int i = 0; i < s; i++) {
            System.out.println(a[i]);
        }
    }
}
```



```
guest@user-user:~$ javac SortingNames.java
guest@user-user:~$ java SortingNames
Enter array size
4
Enter 4 Names
Rahul
Sravan
Vamshi
Arun
Given Names in Sorting order is:

Arun
Rahul
Sravan
Vamshi
```

7. Write a Java program to give an example for command line arguments.

```
class Commandline {
    public static void main(String args[]) {
        if (args.length == 0) {
            System.out.println("No command-line arguments provided.");
        } else {
            System.out.println("Command-line arguments are:");
            for (int i = 0; i < args.length; i++) {
                System.out.println(args[i]);
            } }  } }
```

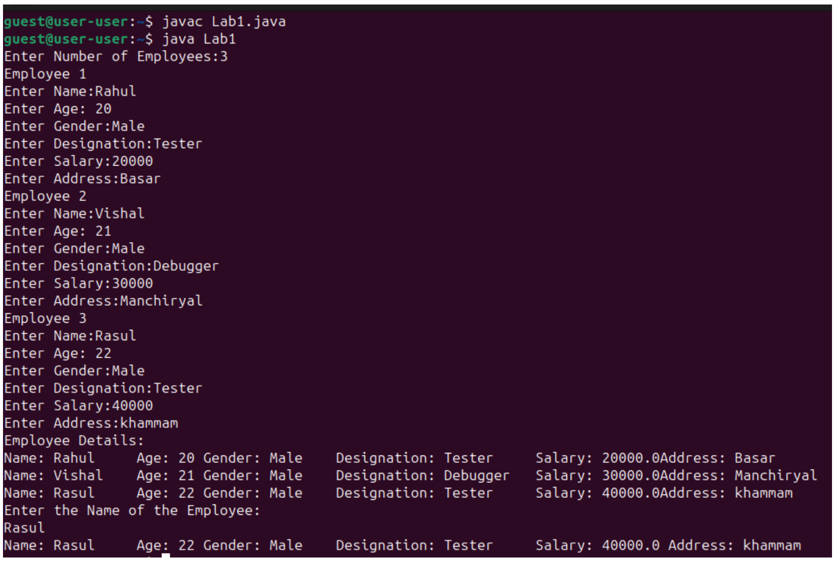


```
guest@user-user:~$ javac Commandline.java
guest@user-user:~$ java Commandline Rahul Arun Pranay Tharun Sravan
Rahul
Arun
Pranay
Tharun
Sravan
```

Week-III

1. Write a program to display details of the required employee based on his Id. The details of employees includes, Emp_name, Emp_age, Emp_gender, Emp_designation, Emp_salary, Emp_Address etc.

```
import java.util.Scanner;

class Employee {
    String Emp_name, Emp_gender, Emp_designation, Emp_address;
    float Emp_salary;
    int Emp_age;

    Employee(String str1, int a, String str2, String str3, float s, String str4) {
        Emp_name = str1;
        Emp_age = a;
        Emp_gender = str2;
        Emp_designation = str3;
        Emp_salary = s;
        Emp_address = str4;
    }

    void getData() {
        System.out.println("Name: " + Emp_name + "\tAge: " + Emp_age + "\tGender: " + Emp_gender +
                "\tDesignation: " + Emp_designation + "\tSalary: " + Emp_salary + "\tAddress: " + Emp_address);
    }
}

class Lab1 {
    public static void main(String... arr) {
        Scanner scan = new Scanner(System.in);
        int size;
        System.out.print("Enter Number of Employees: ");
        size = scan.nextInt();
        scan.nextLine();

        Employee emp[] = new Employee[size];

        for (int i = 0; i < size; i++) {
            System.out.println("Employee " + (i + 1));
            System.out.print("Enter Name: ");
            String name = scan.nextLine();
            System.out.print("Enter Age: ");
            int age = scan.nextInt();
            scan.nextLine();

            System.out.print("Enter Gender: ");
            String gender = scan.nextLine();
            System.out.print("Enter Designation: ");
            String designation = scan.nextLine();
            System.out.print("Enter Salary: ");
            float salary = scan.nextFloat();
            scan.nextLine();
            System.out.print("Enter Address: ");
            String address = scan.nextLine();
```

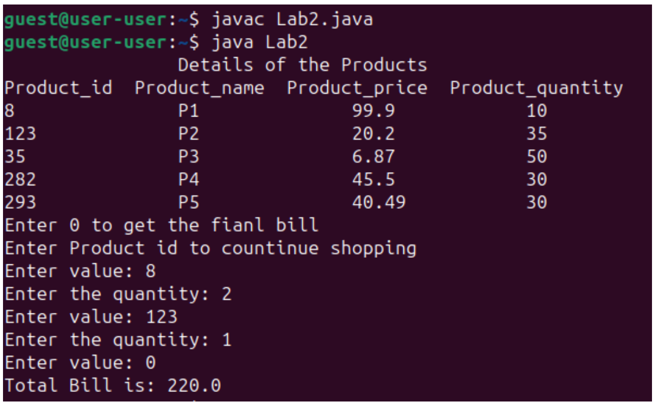


```
guest@user-user:~$ javac Lab1.java
guest@user-user:~$ java Lab1
Enter Number of Employees:3
Employee 1
Enter Name:Rahul
Enter Age: 20
Enter Gender:Male
Enter Designation:Tester
Enter Salary:20000
Enter Address:Basar
Employee 2
Enter Name:Vishal
Enter Age: 21
Enter Gender:Male
Enter Designation:Debugger
Enter Salary:30000
Enter Address:Manchiryal
Employee 3
Enter Name:Rasul
Enter Age: 22
Enter Gender:Male
Enter Designation:Tester
Enter Salary:40000
Enter Address:khammam
Employee Details:
Name: Rahul     Age: 20 Gender: Male    Designation: Tester     Salary: 20000.0Address: Basar
Name: Vishal    Age: 21 Gender: Male    Designation: Debugger   Salary: 30000.0Address: Manchiryal
Name: Rasul     Age: 22 Gender: Male    Designation: Tester     Salary: 40000.0Address: khammam
Enter the Name of the Employee:
Rasul
Name: Rasul     Age: 22 Gender: Male    Designation: Tester     Salary: 40000.0 Address: khammam
```

```
            emp[i] = new Employee(name, age, gender, designation, salary, address);
        }

        System.out.println("Employee Details:");
        for (int i = 0; i < size; i++) {
            emp[i].getData();
        }

        System.out.println("Enter the Name of the Employee to search:");
        String search = scan.nextLine().trim();
        boolean found = false;

        for (int i = 0; i < size; i++) {
            if (search.equalsIgnoreCase(emp[i].Emp_name)) {
                emp[i].getData();
                found = true;
            }
        }

        if (!found) {
            System.out.println("No employee found with the name: " + search);
        }

        scan.close();
    }
}
```

2. A mail-order house sells five products whose retail prices are as follows : Product 1 : Rs. 99.90 , Product 2 : Rs. 20.20 , Product 3 : Rs. 6.87 , Product 4 : Rs. 45.50 and Product 5:Rs. 40.49 . Each product has Prdouct_Id, Product_Name,Product_Quantity, Product_Price. Write an application that reads a series of pairs of numbers as follows :
a. product Id
b. quantity sold. your program should calculate and display the total retail value of all products sold

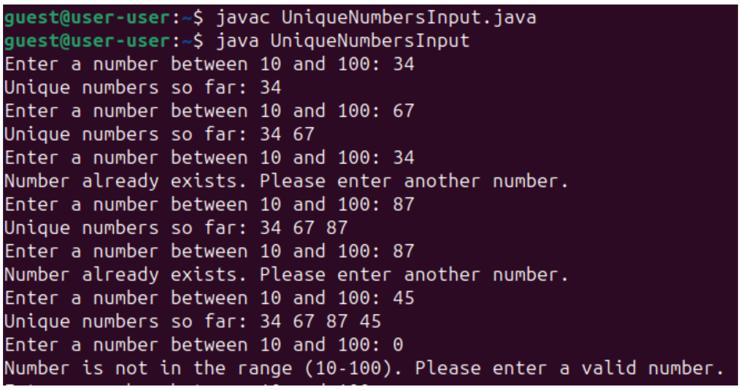
```
import java.lang.*;
import java.util.*;

class Product
{
  String product_name;
  float product_price;
  int product_quantity, product_id;
  Product(int id, String str2, int size, float price)
  {
    product_id = id;
    product_name = str2;
    product_quantity =size;
    product_price =price;
  }
  void displayDetails()
  {
    System.out.println(product_id+"\t\t"+product_name+"\t\t"+product_price+"\t\t"+product_quantity);
  }
}

class Lab2{
  public static void main(String...arr)
  {
   int quantity=0,value;
   float amount=0;
   Scanner scan = new Scanner(System.in);
   Product product[] = new Product[5];
   //for(int i=0; i<5; i++)
   product[0] = new Product(010, "P1", 10, 99.9f);
   product[1] = new Product(123, "P2", 35, 20.2f);
   product[2] = new Product(043, "P3", 50, 6.87f);
   product[3] = new Product(282, "P4", 30, 45.5f);
   product[4] = new Product(293, "P5", 30, 40.49f);
   System.out.println("\t\tDetails of the Products");
   System.out.println("Product_id Product_name Product_price Product_quantity");
   for(int i=0; i<5; i++)
    product[i].displayDetails();

 //Shopping is starts from here
   System.out.println("Enter 0 to get the fianl bill\nEnter Product id to countinue shopping");
   System.out.print("Enter value: ");
   value=scan.nextInt();
   while(value != 0)
   {
    for(int i=0; i<5; i++)
    {
     if(product[i].product_id==value)
     {
     System.out.print("Enter the quantity: ");
     quantity=scan.nextInt();
     if(quantity <= product[i].product_quantity)
     {
      product[i].product_quantity -= quantity;
      amount+= ( (product[i].product_price)*quantity );
     }
     else
     {
      if(product[i].product_quantity == 0)
       System.out.println("Out of stock");
      else
       System.out.println("Insufficient stock");
     }
     break;
    }
   }
   System.out.print("Enter value: ");
   value=scan.nextInt();
  }
  System.out.println("Total Bill is: "+amount);
 }
}
```



```
guest@user-user:~$ javac Lab2.java
guest@user-user:~$ java Lab2
                Details of the Products
Product_id  Product_name  Product_price  Product_quantity
8               P1            99.9           10
123             P2            20.2           35
35              P3            6.87           50
282             P4            45.5           30
293             P5            40.49          30
Enter 0 to get the fianl bill
Enter Product id to countinue shopping
Enter value: 8
Enter the quantity: 2
Enter value: 123
Enter the quantity: 1
Enter value: 0
Total Bill is: 220.0
```

3. Write a java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it’s not a duplicate of any number already read display the complete set of unique values input after the user enters each new value

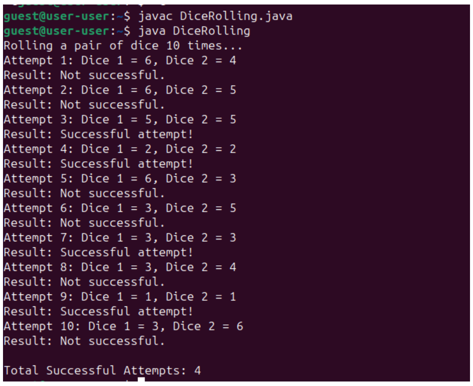
```
import java.util.*;

public class UniqueNumbersInput {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    Set<Integer> uniqueNumbers = new LinkedHashSet<>();

    while (uniqueNumbers.size() < 5) {
      System.out.print("Enter a number between 10 and 100: ");
      int number = scanner.nextInt();

      if (number < 10 || number > 100) {
        System.out.println("Number is not in the range (10-100). Please enter a valid number.");
      } else if (uniqueNumbers.contains(number)) {
        System.out.println("Number already exists. Please enter another number.");
      } else {
        uniqueNumbers.add(number);
        System.out.print("Unique numbers so far: ");
        for (int num : uniqueNumbers) {
          System.out.print(num + " ");
        }
        System.out.println();
      }
    }

    scanner.close();
  }
}
```



```
guest@user-user:~$ javac UniqueNumbersInput.java
guest@user-user:~$ java UniqueNumbersInput
Enter a number between 10 and 100: 34
Unique numbers so far: 34
Enter a number between 10 and 100: 67
Unique numbers so far: 34 67
Enter a number between 10 and 100: 34
Number already exists. Please enter another number.
Enter a number between 10 and 100: 87
Unique numbers so far: 34 67 87
Enter a number between 10 and 100: 87
Number already exists. Please enter another number.
Enter a number between 10 and 100: 45
Unique numbers so far: 34 67 87 45
Enter a number between 10 and 100: 0
Number is not in the range (10-100). Please enter a valid number.
```

4. Write a java program : rolling a pair of dice 10 times [each attempt should be delayed by 10000 ms] and count the number Successful attempts. successful attempt : If the pair of Dice results in the same values.

```
import java.util.Random;

public class DiceRolling {
  public static void main(String[] args) throws InterruptedException {
    Random random = new Random();
    int successfulAttempts = 0;

    System.out.println("Rolling a pair of dice 10 times...");

    for (int i = 1; i <= 10; i++) {
      // Roll two dice
      int dice1 = random.nextInt(6) + 1; // Random number between 1 and 6
      int dice2 = random.nextInt(6) + 1;

      System.out.println("Attempt " + i + ": Dice 1 = " + dice1 + ", Dice 2 = " + dice2);

      // Check if it's a successful attempt
      if (dice1 == dice2) {
        System.out.println("Result: Successful attempt!");
        successfulAttempts++;
      } else {
        System.out.println("Result: Not successful.");
      }
```



```
guest@user-user:~$ javac DiceRolling.java
guest@user-user:~$ java DiceRolling
Rolling a pair of dice 10 times...
Attempt 1: Dice 1 = 6, Dice 2 = 4
Result: Not successful.
Attempt 2: Dice 1 = 6, Dice 2 = 5
Result: Not successful.
Attempt 3: Dice 1 = 5, Dice 2 = 5
Result: Successful attempt!
Attempt 4: Dice 1 = 2, Dice 2 = 2
Result: Successful attempt!
Attempt 5: Dice 1 = 6, Dice 2 = 3
Result: Not successful.
Attempt 6: Dice 1 = 3, Dice 2 = 5
Result: Not successful.
Attempt 7: Dice 1 = 3, Dice 2 = 3
Result: Successful attempt!
Attempt 8: Dice 1 = 3, Dice 2 = 4
Result: Not successful.
Attempt 9: Dice 1 = 1, Dice 2 = 1
Result: Successful attempt!
Attempt 10: Dice 1 = 3, Dice 2 = 6
Result: Not successful.

Total Successful Attempts: 4
```

```
}
```

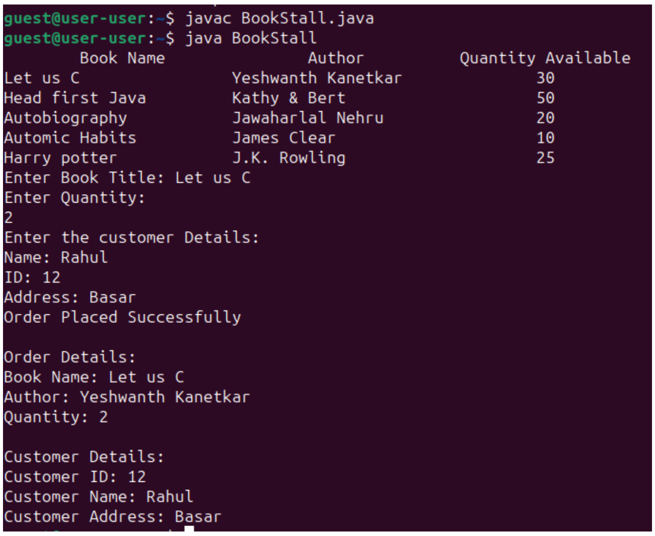
```
        // Delay for 10 seconds
        if (i < 10) { // No need to delay after the last roll
          Thread.sleep(10000);
        }
      }

      System.out.println("\nTotal Successful Attempts: " + successfulAttempts);
  }
}
```

5. Implement the following case study using OOP concepts in Java.E-Book stall : Every book has Properties i.e. Book_Name, Book_Author, Book_Count ; Every Customer has properties as : Customer_Id, Customer_Name,Customer_Address and he can buy Books from the E-Book stall by giving book name, author name and number of books he/she want to buy .Write a Program which will display the list books bought by the customer and remaining text books in the E-book stall with the the count.

```
import java.util.*;
import java.lang.*;

class Book{
  String book_name, book_author;
  int book_count;
  Book(String title, String author, int quantity){
    book_name = title;
    book_author = author;
    book_count = quantity;
  }
  void bookDetails(){
    System.out.println(book_name+"\t\t"+book_author+"\t\t"+book_count);
  }
}

class Customer{
  String customer_id, customer_name, customer_address;
  Customer(String id, String name, String address){
    customer_id = id;
    customer_name = name;
    customer_address = address;
  }
  void customerDetails(){
    System.out.println(customer_id+"\t"+customer_name+"\t"+customer_address);
  }
}

class Market{
  int quantity=0;
  int checkAvailability(String title, Book books[]){
    Scanner scan = new Scanner(System.in);
    int i;
    for(i=0; i<5; i++)
    {

      if( title.equalsIgnoreCase(books[i].book_name) )
      {
        System.out.println("Enter Quantity: ");
        quantity = scan.nextInt();
```

```
 if( quantity > books[i].book_count )
       {
        if(books[i].book_count == 0)
          System.out.println("Insufficient Books available");
        return 0;
       }
       else
       {
        books[i].book_count -= quantity;
        return 1;
       }
     }
   }
   return 0;
  }

  void billing(Book books[], Customer customers, String title)
  {
   System.out.println("Order Placed Successfully\n");
   System.out.println("Order Details:");
   for(int i=0; i<5; i++)
   {
    if( title.equalsIgnoreCase(books[i].book_name) )
         System.out.println("Book  Name:  "+books[i].book_name+"\nAuthor:  "+books[i].book_author+"\nQuantity:  "+
quantity);
   }
   System.out.println("\nCustomer Details:");
                                   System.out.println("Customer        ID:        "+customers.customer_id+"\nCustomer        Name:
"+customers.customer_name+"\nCustomer Address: "+customers.customer_address);
  }
}

class BookStall{
  public static void main(String... arr){
   //Variables
   int check,quantity;
   String name, id, address, title = new String();
   //Objects
   Scanner scan = new Scanner(System.in);
   Market business = new Market();
   Book listOfBooks[] = new Book[5];
   //List of Books Available
   listOfBooks[0] = new Book("Let us C","Yeshwanth Kanetkar",30);
   listOfBooks[1] = new Book("Head first Java","Kathy & Bert   ",50);
   listOfBooks[2] = new Book("Autobiography","Jawaharlal Nehru ",20);
   listOfBooks[3] = new Book("Automic Habits","James Clear   ",10);
   listOfBooks[4] = new Book("Harry potter","J.K. Rowling   ",25);
   //Displaying the List of Books available in the book stall
   System.out.println("\tBook Name\t\tAuthor\t\tQuantity Available");
   for(int i=0 ; i<5 ; i++)
    listOfBooks[i].bookDetails();
   //Transaction
   System.out.print("Enter Book Title: ");
   title = scan.nextLine();
   check = business.checkAvailability(title,listOfBooks);
   if(check == 0)
   {
    System.out.println("Sorry "+title+" Book is not available\nThank you for visiting!!!");
```

```
      System.exit(0);
    }
    //Customer details
    System.out.println("Enter the customer Details:");
    System.out.print("Name: ");
    name = scan.nextLine();
    System.out.print("ID: ");
    id = scan.nextLine();
    System.out.print("Address: ");
    address = scan.nextLine();
    Customer buyer = new Customer(id,name,address);

    //Billing
    business.billing(listOfBooks,buyer,title);
  }
}
```

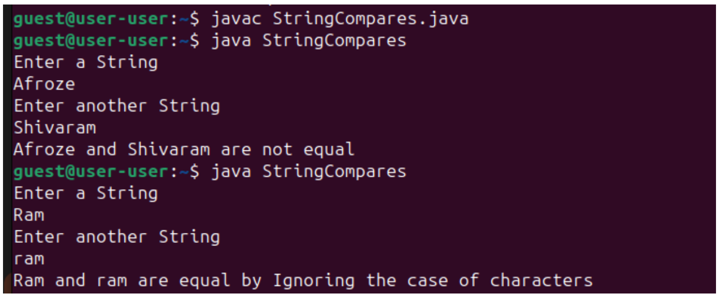


```
guest@user-user:~$ javac BookStall.java
guest@user-user:~$ java BookStall
        Book Name               Author          Quantity Available
Let us C                Yeshwanth Kanetkar              30
Head first Java         Kathy & Bert                    50
Autobiography           Jawaharlal Nehru                20
Automic Habits          James Clear                     10
Harry potter            J.K. Rowling                    25
Enter Book Title: Let us C
Enter Quantity:
2
Enter the customer Details:
Name: Rahul
ID: 12
Address: Basar
Order Placed Successfully

Order Details:
Book Name: Let us C
Author: Yeshwanth Kanetkar
Quantity: 2

Customer Details:
Customer ID: 12
Customer Name: Rahul
Customer Address: Basar
```

Week-IV

1. Write an application that uses the String method “compareTo” to compare two strings defined by the user.

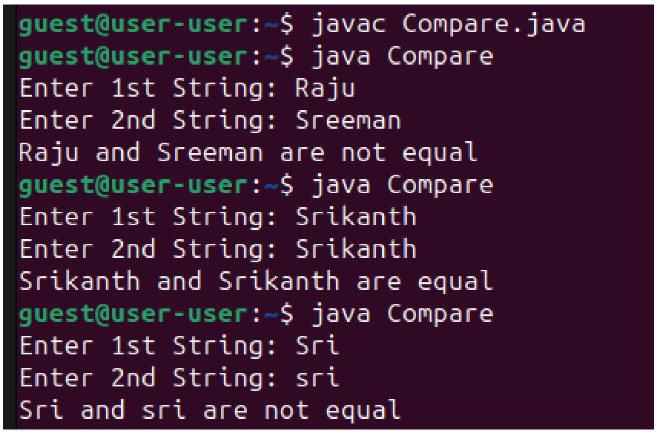
```
import java.util.Scanner;

public class Compare {
  public static void main (String...arr) {
    String str1, str2;
    Scanner scan = new Scanner (System.in);
    System.out.print("Enter 1st String: ");
    str1 = scan.nextLine();
    System.out.print("Enter 2nd String: ");
    str2 = scan.nextLine();
    if (str1.compareTo(str2) == 0)
      System.out.println(str1+" and "+str2+" are equal");
    else
      System.out.println(str1+" and "+str2+" are not equal");
    scan.close();
  }
}
```



```
guest@user-user:~$ javac StringCompares.java
guest@user-user:~$ java StringCompares
Enter a String
Afroze
Enter another String
Shivaram
Afroze and Shivaram are not equal
guest@user-user:~$ java StringCompares
Enter a String
Ram
Enter another String
ram
Ram and ram are equal by Ignoring the case of characters
```

2. Write an application that uses the String method equals and equalsIgnoreCase to test any two string objects for equality.

```
import java.util.Scanner;

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

        System.out.println("Enter a String:");
        String s = sc.nextLine();

        System.out.println("Enter another String:");
        String s1 = sc.nextLine();

        if (s.equals(s1)) {
            System.out.println(s + " and " + s1 + " are equal");
        } else if (s.equalsIgnoreCase(s1)) {
            System.out.println(s + " and " + s1 + " are equal by ignoring the case of characters");
        } else {
```

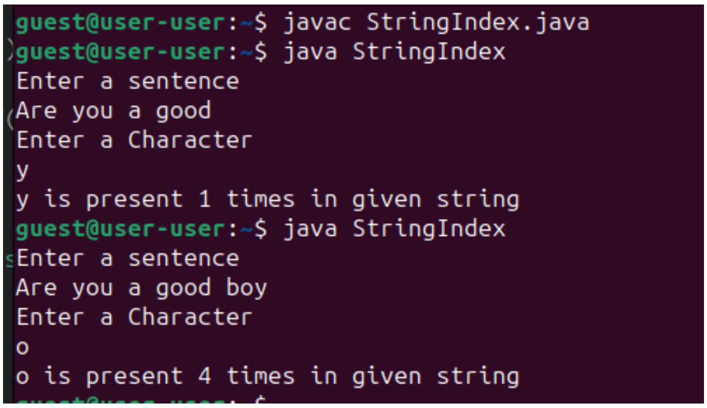


```
guest@user-user:~$ javac Compare.java
guest@user-user:~$ java Compare
Enter 1st String: Raju
Enter 2nd String: Sreeman
Raju and Sreeman are not equal
guest@user-user:~$ java Compare
Enter 1st String: Srikanth
Enter 2nd String: Srikanth
Srikanth and Srikanth are equal
guest@user-user:~$ java Compare
Enter 1st String: Sri
Enter 2nd String: sri
Sri and sri are not equal
```

```
            System.out.println(s + " and " + s1 + " are not equal");
        }

        sc.close();
    }
}
```

3. Write an application that uses String method indexOf to determine the total number of occurrences of any given alphabet in a defined text.

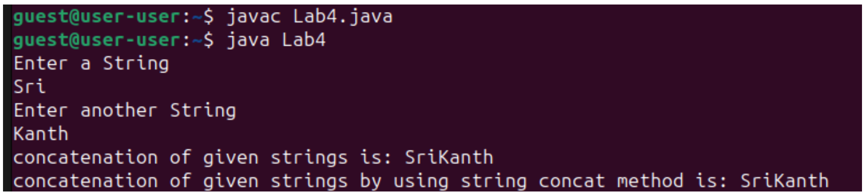
```
import java.util.Scanner;

class StringIndex {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a sentence:");
        String sen = sc.nextLine();
        System.out.println("Enter a character:");
        char ch = sc.next().charAt(0);

        int count = 0;
        for (int i = 0; i < sen.length(); i++) {
            if (sen.charAt(i) == ch) {
                count++;
            }
        }

        System.out.println(ch + " is present " + count + " times in the given string");
        sc.close();
    }
}
```



```
guest@user-user:~$ javac StringIndex.java
guest@user-user:~$ java StringIndex
Enter a sentence
Are you a good
Enter a Character
y
y is present 1 times in given string
guest@user-user:~$ java StringIndex
Enter a sentence
Are you a good boy
Enter a Character
o
o is present 4 times in given string
```

4. Write an application that uses String method concat to concatenate two defined strings.

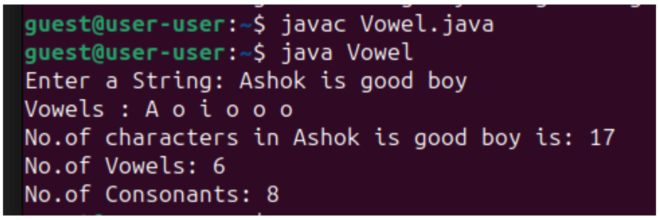
```
import java.util.Scanner;

class Week4 {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a String:");
        String s = sc.nextLine();

        System.out.println("Enter another String:");
        String s1 = sc.nextLine();

        System.out.println("Concatenation of given strings is: " + s + s1);

        // Concatenating using the concat() method
        String s2 = s.concat(s1);
        System.out.println("Concatenation of given strings by using String concat method is: " + s2);

        sc.close();
    }
}
```



```
guest@user-user:~$ javac Lab4.java
guest@user-user:~$ java Lab4
Enter a String
Sri
Enter another String
Kanth
concatenation of given strings is: SriKanth
concatenation of given strings by using string concat method is: SriKanth
```

5. Write a Java program to print all vowels in given string and count number of vowels and consonants present in given string

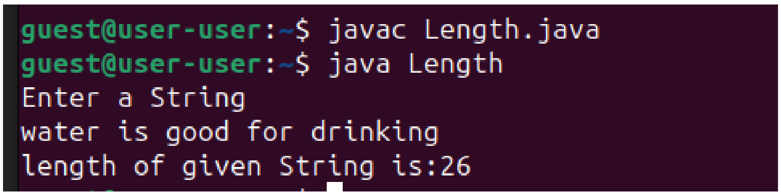
```
import java.util.Scanner;

public class Vowel{
  public static void main (String...arr) {
    String str;
    Scanner scan = new Scanner (System.in);
    System.out.print("Enter a String: ");
    str = scan.nextLine();
    findCount(str);
    scan.close();
  }
  static void findCount(String s) {
    int vowelcount = 0, consocount = 0, size = s.length();
    System.out.print("Vowels : ");
    for(char c: s.toCharArray())
    {
      if( (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') ||
        (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U') ) {
          System.out.print(c+" ");
          vowelcount++;
        }
      else if(( c >= 'a' && c <= 'z') || ( c >= 'A' && c <= 'Z'))
        consocount++;
      else;
    }
    System.out.println("\nNo.of characters in "+s+" is: "+size);
    System.out.println("No.of Vowels: "+vowelcount);
    System.out.println("No.of Consonants: "+consocount);
  }
}
```



```
guest@user-user:~$ javac Vowel.java
guest@user-user:~$ java Vowel
Enter a String: Ashok is good boy
Vowels : A o i o o o
No.of characters in Ashok is good boy is: 17
No.of Vowels: 6
No.of Consonants: 8
```

6. Write an application that finds the length of a given string.

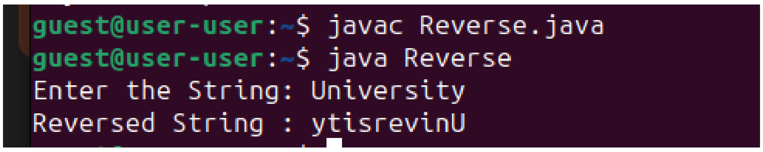
```
import java.util.Scanner;

class Length {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a String");
        String s = sc.nextLine();

        int a = s.length();
        System.out.println("Length of given String is: " + a);

        sc.close(); // Closing the scanner
    }
}
```



```
guest@user-user:~$ javac Length.java
guest@user-user:~$ java Length
Enter a String
water is good for drinking
length of given String is:26
```

7. Write an application that uses the String method charAt to reverse the string.

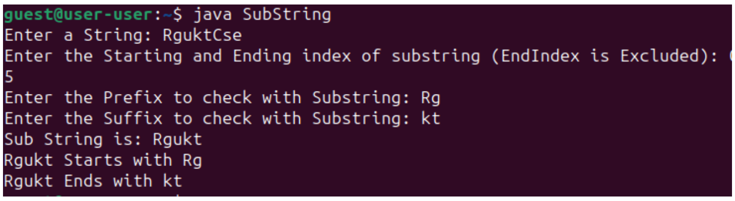
```
import java.util.*;
public class Reverse {
  public static void main (String...arr) {
    Scanner scan = new Scanner(System.in);
    String str,reverse;
    System.out.print("Enter the String: ");
    str = scan.nextLine();
    reverse = reveresOf(str);
    System.out.println("Reversed String : "+reverse);
    scan.close();
  }
  static String reveresOf(String s) {
    String result = "";
    for(int i = s.length()-1 ; i >= 0 ; i--) {
      result += s.charAt(i);
    }
    return result;
  }
}
```



```
guest@user-user:~$ javac Reverse.java
guest@user-user:~$ java Reverse
Enter the String: University
Reversed String : ytisrevinU
```

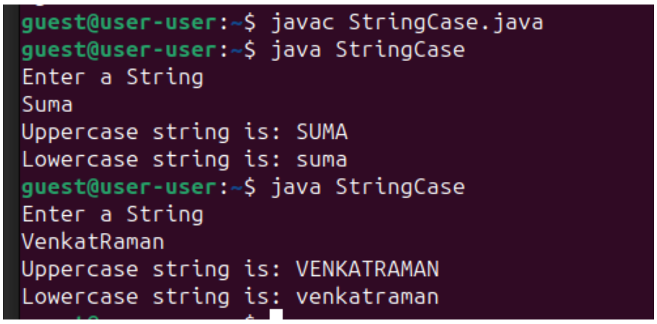
8. Write an application that finds the substring from any given string using substring method and startsWith & endsWith methods.

```
import java.util.Scanner;

public class SubString{
  public static void main (String...arr) {
    String originalStr, prefix, suffix, resultstring;
    int start, end;
    Scanner scan = new Scanner(System.in);
    System.out.print("Enter a String: ");
    originalStr = scan.nextLine();
    System.out.print("Enter the Starting and Ending index of substring (EndIndex is Excluded): ");
    start = scan.nextInt();
    end = scan.nextInt();
    scan.nextLine();
    System.out.print("Enter the Prefix to check with Substring: ");
    prefix = scan.nextLine();
    System.out.print("Enter the Suffix to check with Substring: ");
    suffix = scan.nextLine();
    resultstring = originalStr.substring(start,end);
    System.out.println("Sub String is: "+ resultstring);
    if(resultstring.startsWith(prefix))
      System.out.println(resultstring+" Starts with "+prefix);
    else
      System.out.println(resultstring+" is not Starts with "+prefix);
    if(resultstring.endsWith(suffix))
      System.out.println(resultstring+" Ends with "+suffix);
    else
      System.out.println(resultstring+" is not Ends with "+suffix);

  }
}
```



```
guest@user-user:~$ java SubString
Enter a String: RguktCse
Enter the Starting and Ending index of substring (EndIndex is Excluded):
5
Enter the Prefix to check with Substring: Rg
Enter the Suffix to check with Substring: kt
Sub String is: Rgukt
Rgukt Starts with Rg
Rgukt Ends with kt
```

9. Write an application that changes any given string with uppercase letters, displays it, changes it back to lowercase letters and displays it.



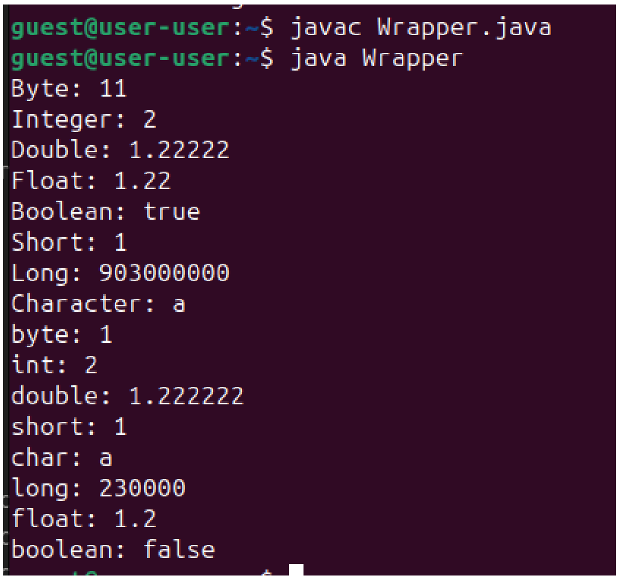
```
guest@user-user:~$ javac StringCase.java
guest@user-user:~$ java StringCase
Enter a String
Suma
Uppercase string is: SUMA
Lowercase string is: suma
guest@user-user:~$ java StringCase
Enter a String
VenkatRaman
Uppercase string is: VENKATRAMAN
Lowercase string is: venkatraman
```

```
import java.util.Scanner;

class StringCase {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a String");
        String s = sc.nextLine();

        String uppercaseString = s.toUpperCase();
        System.out.println("Uppercase string is: " + uppercaseString);

        String lowercaseString = uppercaseString.toLowerCase();
        System.out.println("Lowercase string is: " + lowercaseString);

        sc.close(); // Close the scanner to avoid resource leak
    }
}
```

Week-V

1. Write a Java Program to implement Wrapper classes and their methods.

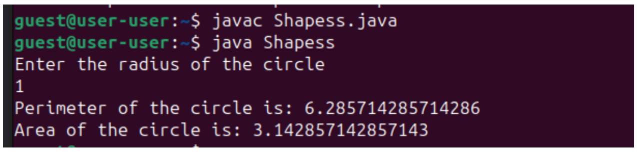
```
package lab;
import java.util.*;
class Autoboxing{
  Autoboxing(byte a){
   Byte b=a;
 System.out.println("Byte: "+b);
  }
  Autoboxing(int a){
   Integer b=a;
   System.out.println("Integer: "+b);
  }
  Autoboxing(double a){
   Double b=a;
   System.out.println("Double: "+b);
  }
  Autoboxing(float a){
   Float b=a;
   System.out.println("Float: "+b);
  }
  Autoboxing(boolean a){
   Boolean b=a;
   System.out.println("Boolean: "+b);
  }
  Autoboxing(short a){
   Short b=a;
   System.out.println("Short: "+b);
  }
  Autoboxing(long a){
   Long b=a;
   System.out.println("Long: "+b);
  }
  Autoboxing(char a){
   Character b=a;
   System.out.println("Character: "+b);
  }

}
class Unboxing{
static void unboxing(Byte A,Integer I,Double D,Short S,Character C,Long L,Float F,Boolean BL){

 byte b=A.byteValue();
  System.out.println("byte: "+b);
  int i=I.intValue();
  System.out.println("int: "+i);
  double d=D.doubleValue();
  System.out.println("double: "+d);
  short s=S.shortValue();
  System.out.println("short: "+s);
  char c=C.charValue();
  System.out.println("char: "+c);
  long l=L.longValue();
  System.out.println("long: "+l);
  float f=F.floatValue();
  System.out.println("float: "+f);
  boolean bl=BL.booleanValue();
  System.out.println("boolean: "+bl);
 }
 }
 public class Lab51 {
  public static void main(String[] args) {
  byte b=11;
  int i=2;
  double d=1.22222;
  short s=1;
  char c='a';
  long l=903000000;
  float f=1.22f;
  boolean bl=true;
  Autoboxing ob1=new Autoboxing(b);
  Autoboxing ob2=new Autoboxing(i);
  Autoboxing ob3=new Autoboxing(d);
  Autoboxing ob4=new Autoboxing(f);
  Autoboxing ob5=new Autoboxing(bl);
  Autoboxing ob6=new Autoboxing(s);
  Autoboxing ob7=new Autoboxing(l);
  Autoboxing ob8=new Autoboxing(c);
  Byte A=1;
  Integer I=2;
  Double D=1.222222;
  Short S=1;
  Character C='a';
  Float F=1.2f;
  Long L=230000l;
  Boolean BL=false;
  Unboxing.unboxing(A,I,D,S,C,L,F,BL);
  }
 }
```



```
guest@user-user:~$ javac Wrapper.java
guest@user-user:~$ java Wrapper
Byte: 11
Integer: 2
Double: 1.22222
Float: 1.22
Boolean: true
Short: 1
Long: 903000000
Character: a
byte: 1
int: 2
double: 1.222222
short: 1
char: a
long: 230000
float: 1.2
boolean: false
```

2. Write an application that prompts the user for the radius of a circle and uses a method called circleArea to calculate the area of the circle and uses a method circlePerimeter to calculate the perimeter of the circle.

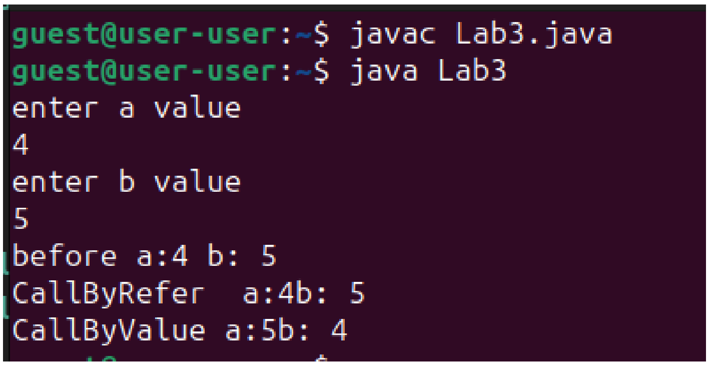
```
import java.util.Scanner;

class Circle {
    double r;

    double Area() {
        return (22.0 / 7) * r * r;
    }

    double Perimeter() {
        return 2 * (22.0 / 7) * r;
    }
}

class Demo {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        Circle c = new Circle(); // object of Circle class
        System.out.println("Enter the radius of the circle");
        double R = s.nextDouble();
        c.r = R;
        System.out.println("Perimeter of the circle is: " + c.Perimeter());
        System.out.println("Area of the circle is: " + c.Area());
    }
}
```



```
guest@user-user:~$ javac Shapess.java
guest@user-user:~$ java Shapess
Enter the radius of the circle
1
Perimeter of the circle is: 6.285714285714286
Area of the circle is: 3.142857142857143
```

3. Write a JAVA program for the following.
a. Call by value
b. Call by object

```
import java.util.*;
class CallByValue{
  static void swap(Number a1,Number b1){
    int t=a1.value;
    a1.value=b1.value;
    b1.value=t;
  }

}
class CallByRefer{
  static void swap(int a,int b) {
    int t=a;
    a=b;
    b=t;
  }

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

public class Lab53 {
public static void main(String[] args) {
  int a,b;
  Scanner scan=new Scanner(System.in);
  System.out.println("enter a value");
  a=scan.nextInt();
  System.out.println("enter b value");
  b=scan.nextInt();
  Number a1=new Number(a);
  Number b1=new Number(b);
  System.out.println("before a:"+a+" b: "+b);
  CallByRefer.swap(a,b);
  System.out.println("CallByRefer a:"+a+"b: "+b);
  CallByValue.swap(a1,b1);
  System.out.println("CallByValue a:"+a1.value+"b: "+b1.value);

 }
 }
```

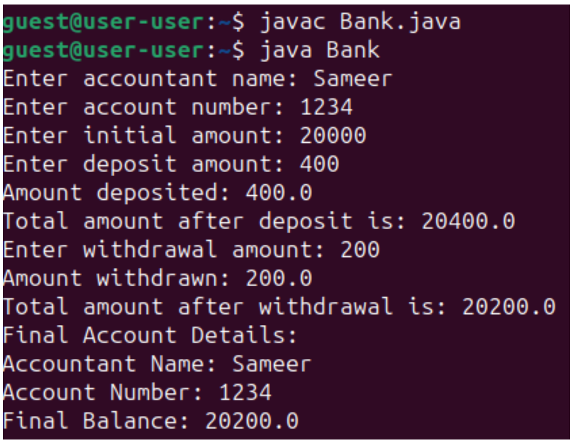


```
guest@user-user:~$ javac Lab3.java
guest@user-user:~$ java Lab3
enter a value
4
enter b value
5
before a:4 b: 5
CallByRefer  a:4b: 5
CallByValue a:5b: 4
```

4. Create a class Account with an instance variable balance (double). It should contain a constructor that initializes the balance, ensuring that the initial balance is greater than 0.0. Acct details: Acct_Name, Acct_acctno, Acct_Bal, Acct_Address.Create two methods namely credit and debit, getBalance. The Credit adds the amount (passed as parameter) to balance and does not return any data. Debit method withdraws money from an Account. GetBalance displays the amount. Ensure that the debit amount does not exceed the Account's balance. In that case the balance should be left unchanged and the method should print a message indicating"Debit amount exceeded account balance".

```
import java.util.Scanner;

class Account {

    String accountantName;
    int accountNumber;
    double amount;

    Account(String name, double initialAmount, int number) {
        accountantName = name;
        amount = initialAmount;
        accountNumber = number;
    }

    void debit(double withdrawAmount) {
        if (amount >= withdrawAmount) {
            amount -= withdrawAmount;
            System.out.println("Amount withdrawn: " + withdrawAmount);
            System.out.println("Total amount after withdrawal is: " + amount);
        } else {
            System.out.println("Insufficient account balance.");
            System.out.println("Current balance: " + amount);
        }
    }

    void credit(double depositAmount) {
        amount += depositAmount;
        System.out.println("Amount deposited: " + depositAmount);
        System.out.println("Total amount after deposit is: " + amount);
    }
}
```

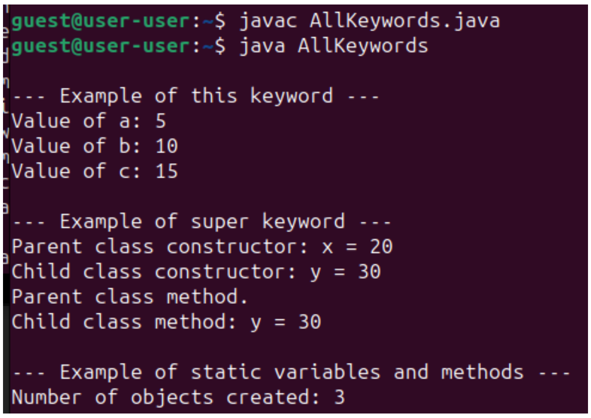
```
class Bank {
 public static void main(String args[]) {
 Scanner s = new Scanner(System.in);

 System.out.print("Enter accountant name: ");
 String name = s.nextLine();

 System.out.print("Enter account number: ");
 int accountNumber = s.nextInt();

 System.out.print("Enter initial amount: ");
 double initialAmount = s.nextDouble();

 Account account = new Account(name, initialAmount, accountNumber);

 System.out.print("Enter deposit amount: ");
 double depositAmount = s.nextDouble();
 account.credit(depositAmount);

 System.out.print("Enter withdrawal amount: ");
 double withdrawAmount = s.nextDouble();
 account.debit(withdrawAmount);

 System.out.println("Final Account Details:");
 System.out.println("Accountant Name: " + account.accountantName);
 System.out.println("Account Number: " + account.accountNumber);
 System.out.println("Final Balance: " + account.amount);
 }
}
```



```
guest@user-user:~$ javac Bank.java
guest@user-user:~$ java Bank
Enter accountant name: Sameer
Enter account number: 1234
Enter initial amount: 20000
Enter deposit amount: 400
Amount deposited: 400.0
Total amount after deposit is: 20400.0
Enter withdrawal amount: 200
Amount withdrawn: 200.0
Total amount after withdrawal is: 20200.0
Final Account Details:
Accountant Name: Sameer
Account Number: 1234
Final Balance: 20200.0
```

5. Write Java program for the following
a. Example for this operator and the use of this keyword.
b. Example for super keywords.
c. Example for static variables and methods.

```
class Parent {
    int x;

    Parent(int x) {
        this.x = x; // Using this to reference the instance variable
        System.out.println("Parent class constructor: x = " + x);
    }

    void display() {
        System.out.println("Parent class method.");
    }
}

class Child extends Parent {
    int y;

    Child(int x, int y) {
        super(x); // Calling parent class constructor
        this.y = y; // Using this to reference the instance variable
        System.out.println("Child class constructor: y = " + y);
    }
    @Override
    void display() {
        super.display(); // Calling parent class method
        System.out.println("Child class method: y = " + y);
    }
}

class ThisExample {
    int a, b, c;

    ThisExample(int a, int b, int c) {
        this(a, b); // Calling another constructor in the same class
        this.c = c; // Using this to assign the instance variable
        System.out.println("Value of c: " + this.c);
    }

    ThisExample(int a, int b) {
        this.a = a; // Using this to assign the instance variable
        this.b = b; // Using this to assign the instance variable
        System.out.println("Value of a: " + this.a);

System.out.println("Value of b: " + this.b);
 }
}

class StaticExample {
 static int count = 0; // Static variable

 StaticExample() {
 count++; // Incrementing static variable
 }

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

public class AllInOneDemo {
 public static void main(String[] args) {
 // Example of this keyword
 System.out.println("\n--- Example of this keyword ---");
 ThisExample thisObj = new ThisExample(5, 10, 15);

 // Example of super keyword
 System.out.println("\n--- Example of super keyword ---");
 Child childObj = new Child(20, 30);
 childObj.display();

 // Example of static variables and methods
 System.out.println("\n--- Example of static variables and methods ---");
 StaticExample obj1 = new StaticExample();
 StaticExample obj2 = new StaticExample();
 StaticExample obj3 = new StaticExample();
 StaticExample.displayCount();
 }
}
```

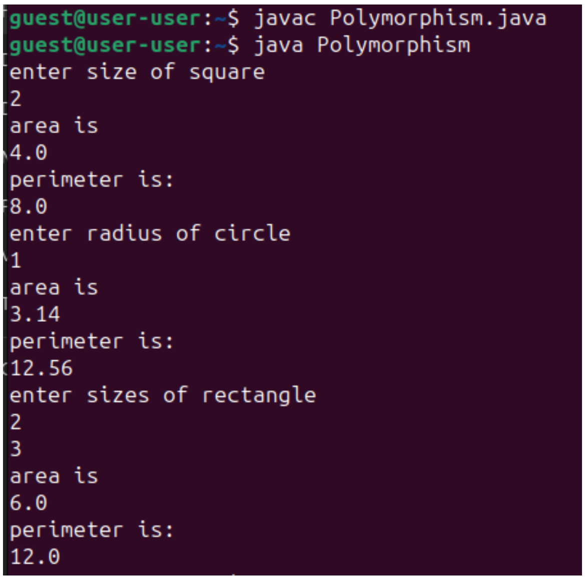


```
guest@user-user:~$ javac AllKeywords.java
guest@user-user:~$ java AllKeywords

--- Example of this keyword ---
Value of a: 5
Value of b: 10
Value of c: 15

--- Example of super keyword ---
Parent class constructor: x = 20
Child class constructor: y = 30
Parent class method.
Child class method: y = 30

--- Example of static variables and methods ---
Number of objects created: 3
```

Week-VI

1. Write a Java program to find Area and Circle of different shapes using polymorphism concept

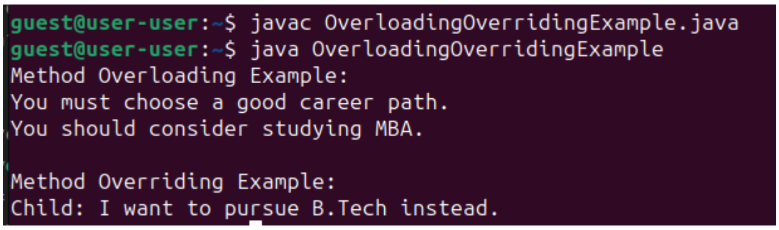
```
package lab;
import java.util.*;
class Shapes{
  final float PIE=3.14f;
  float area,perimeter;
  void area(float length,float breadth){
    System.out.println("area is ");
  }
  void perimeter(float length,float breadth){
   System.out.println("perimeter is: ");
  }
   void area(float length){
    System.out.println("area is ");
  }
  void perimeter(float length){
   System.out.println("perimeter is: ");
  }
  }
class Square extends Shapes{
  void area(float length){
    area=length*length;
    super.area(length);
    System.out.println(area);
  }
  void perimeter(float length){
    perimeter=4*length;
    super.perimeter(length);
   System.out.println(perimeter);
  }
}
class Circle extends Shapes{
  void area(float length){
    area=PIE*length*length;
    super.area(length);
    System.out.println(area);
  }
  void perimeter(float length){
    perimeter=4*length*PIE;
    super.perimeter(length);
   System.out.println(perimeter);
  }
}
class Rectangle extends Shapes{
 void area(float length,float breadth){
 area=length*breadth;
 super.area(length, breadth);
 System.out.println(area);
 }
 void perimeter(float length,float breadth){
 perimeter=2*length*breadth;
 super.perimeter(length, breadth);
 System.out.println(perimeter);
 }
}
public class Lab61 {
 public static void main(String[] args) {
 float length,breadth;
 Scanner scan=new Scanner(System.in);
 System.out.println("enter size of square");
 length=scan.nextFloat();
 Square s=new Square();
 s.area(length);
 s.perimeter(length);
 System.out.println("enter radius of circle");
 length=scan.nextFloat();
 Circle c=new Circle();
 c.area(length);
 c.perimeter(length);
 System.out.println("enter sizes of rectangle");
 length=scan.nextFloat();
 breadth=scan.nextFloat();
 Rectangle r=new Rectangle();
 r.area(length,breadth);
 r.perimeter(length,breadth);
 }

}
```



```
guest@user-user:~$ javac Polymorphism.java
guest@user-user:~$ java Polymorphism
enter size of square
2
area is
4.0
perimeter is:
8.0
enter radius of circle
1
area is
3.14
perimeter is:
12.56
enter sizes of rectangle
2
3
area is
6.0
perimeter is:
12.0
```

2. Write a Java program which can give example of Method overloading and overriding

```
// Parent class
class Parent {
    // Method Overloading in Parent class
    void advice() {
        System.out.println("You must choose a good career path.");
    }

    void advice(String field) {
        System.out.println("You should consider studying " + field + ".");
    }

    // Method to be overridden
    void suggestCareer() {
        System.out.println("Parent: MBA is a good career choice.");
    }
}

// Child class overriding methods from Parent
class Child extends Parent {
    // Overriding the suggestCareer method
    @Override
    void suggestCareer() {
        System.out.println("Child: I want to pursue B.Tech instead.");
    }
}

public class OverloadingOverridingExample {
    public static void main(String[] args) {
        // Demonstrate Method Overloading
```

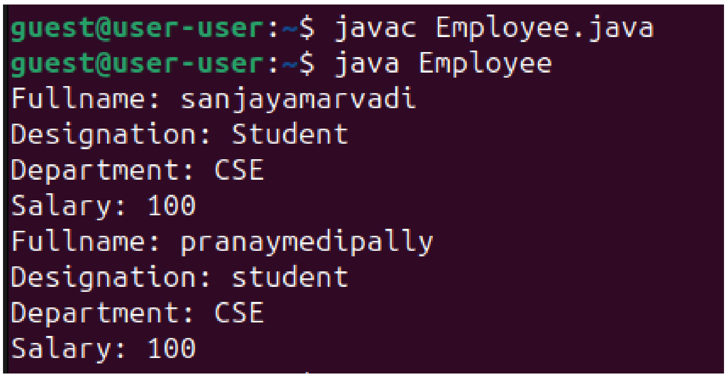


```
guest@user-user:~$ javac OverloadingOverridingExample.java
guest@user-user:~$ java OverloadingOverridingExample
Method Overloading Example:
You must choose a good career path.
You should consider studying MBA.

Method Overriding Example:
Child: I want to pursue B.Tech instead.
```

```
        Parent parent = new Parent();
        System.out.println("Method Overloading Example:");
        parent.advice();
        parent.advice("MBA");

        // Demonstrate Method Overriding
        Child child = new Child();
        System.out.println("\nMethod Overriding Example:");
        child.suggestCareer(); // Calls overridden method in Child
    }
}
```

3. Write an application to create a super class Employee with information first name & last name and methods getFirstName(), getLastName() derive the subclasses ContractEmployee and RegularEmployee with the information about department, designation & method displayFullName() , getDepartment(), getDesig() to print the salary and to set department name & designation of the corresponding sub-class objects respectively.

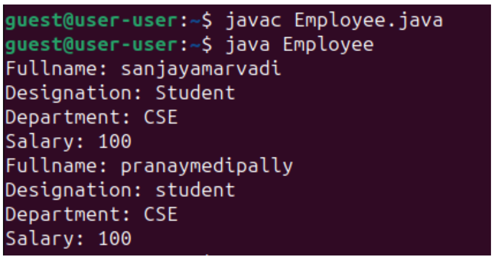
```
package lab;
import java.lang.*;
class Employees{
  String firstname,lastname;
  Employees(String firstname,String lastname){
    this.firstname=firstname;
    this.lastname=lastname;
  }
  void getFirstName(){
    System.out.println("Firstname: "+firstname);
  }
  void getLaastName(){
    System.out.println("Firstname: "+lastname);
  }
}
class ContractEmployee extends Employees{
  String department,designation;
  int salary;
  ContractEmployee(String firstname,String lastname,String department,String designation,int salary){
    super(firstname,lastname);
    this.department=department;
    this.designation=designation;
    this.salary=salary;
  }
    void displayFullname(){
      System.out.println("Fullname: "+super.firstname+super.lastname);
    }
    void getDesignation(){
      System.out.println("Designation: "+designation);
    }
    void getDepartment(){
      System.out.println("Department: "+department);
    }
    void getSalary(){
      System.out.println("Salary: "+salary);
    }
    void display(){
      displayFullname();
      getDesignation();
      getDepartment();
  getSalary();
  }
  }
 class RegularEmployee extends Employees{
  String department,designation;
  int salary;
  RegularEmployee(String firstname,String lastname,String department,String designation,int salary){
  super(firstname,lastname);
  this.department=department;
  this.designation=designation;
  this.salary=salary;
  }
  void displayFullname(){
  System.out.println("Fullname: "+super.firstname+super.lastname);
  }
  void getDesignation(){
  System.out.println("Designation: "+designation);
  }
  void getDepartment(){
  System.out.println("Department: "+department);
  }
  void getSalary(){
  System.out.println("Salary: "+salary);
  }
  void display(){
  displayFullname();
  getDesignation();
  getDepartment();
  getSalary();
  }
  }
 public class Lab63 {
  public static void main(String[] args) {
  ContractEmployee c=new ContractEmployee("sanjay","amarvadi", "CSE","Student",100);
  RegularEmployee r=new RegularEmployee("pranay","medipally", "CSE", "student",100);
  c.display();
  r.display();
  }

 }
```



```
guest@user-user:~$ javac Employee.java
guest@user-user:~$ java Employee
Fullname: sanjayamarvadi
Designation: Student
Department: CSE
Salary: 100
Fullname: pranaymedipally
Designation: student
Department: CSE
Salary: 100
```

4. Derive sub-classes of ContractEmployee namely HourlyEmployee & WeeklyEmployee with information number of hours & wages per hour, number of weeks & wages per week respectively & method calculateWages() to calculate their monthly salary. Also override getDesig () method depending on the type of contract employee.

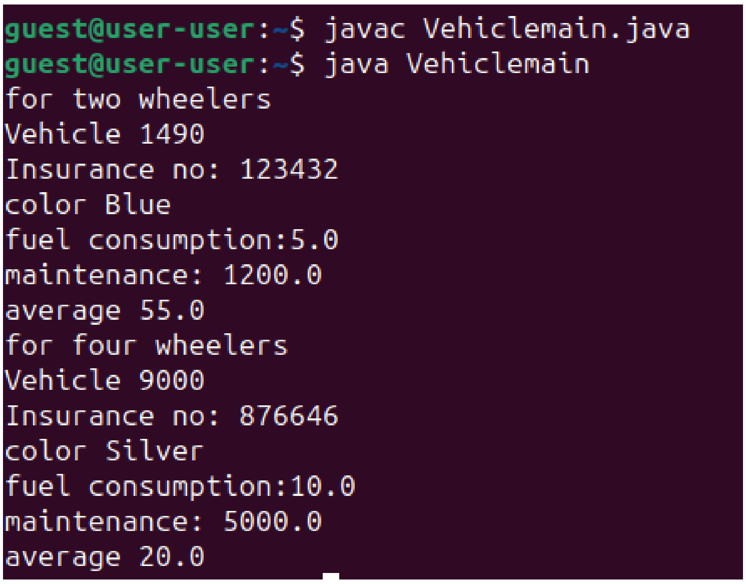
```
package lab;
class HourlyEmployee extends ContractEmployee{
 int hours;
 int hourwage;
 int hoursalary;
   HourlyEmployee(String firstname,String lastname,String department,String designation,int salary,int hours,int hourwage){
   super(firstname,lastname,department,designation,salary);
   this.hours=hours;
   this.hourwage=hourwage;
 }
 void calculateWages(){
     weeksalary=weeks*weekwage;
     display();
     System.out.println("weeksalary: "+weeksalary);
    }

  }

  public class Lab64 {
   public static void main(String[] args) {
   HourlyEmployee h=new HourlyEmployee("sanjay", "amarvadi", "CSE", "student",400,2,10);
   h.calculateWages();
   WeeklyEmployee w=new WeeklyEmployee("pranay", "medipally", "CSE", "student", 1000, 3,5000);
  w.calculateWages();
  }
 }
```



```
guest@user-user:~$ javac Employee.java
guest@user-user:~$ java Employee
Fullname: sanjayamarvadi
Designation: Student
Department: CSE
Salary: 100
Fullname: pranaymedipally
Designation: student
Department: CSE
Salary: 100
```

5. Write an application to create a superclass Vehicle with information vehicle number,insurance number,color and methods getConsumption() displayConsumption(). Derive the sub-classes TwoWheeler and FourWheeler with method maintenance() and average() to print the maintenance And average of the vehicle.

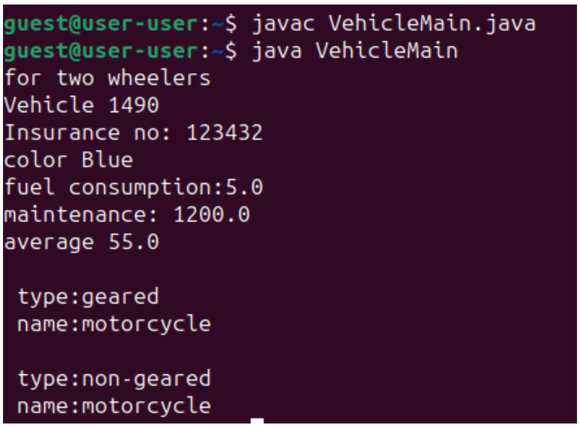
```
class Vehicle {
    int vno;
    int ino;
    String color;
    double fuel;

    void setInfo(int vno, int ino, String color) {
        this.vno = vno;
        this.ino = ino;
        this.color = color;
    }

    void getConsumption(double fuel) {
        this.fuel = fuel;
    }

    void displayConsumption() {
        System.out.println("Fuel consumption: " + fuel);
    }

    void displayInfo() {
        System.out.println("Vehicle number: " + vno);
        System.out.println("Insurance number: " + ino);
        System.out.println("Color: " + color);
    }
}

class TwoWheeler extends Vehicle {
    double avg;
    double mt;

    void setSpecs(double avg, double mt) {
        this.avg = avg;
        this.mt = mt;
    }

    double getMaintenance() {
        return mt;
    }
    double getAverage() {
        return avg;
    }
}

class FourWheeler extends Vehicle {
    double avg;
    double mt;

    void setSpecs(double avg, double mt) {
        this.avg = avg;
        this.mt = mt;
    }

    double getMaintenance() {
        return mt;
    }

    double getAverage() {
        return avg;
    }
}

class VehicleMain {
    public static void main(String[] args) {
        TwoWheeler tw = new TwoWheeler();
        tw.setInfo(1490, 123432, "Blue");
        tw.getConsumption(5);
        tw.setSpecs(55, 1200);
        System.out.println("For Two Wheelers:");
        tw.displayInfo();
        tw.displayConsumption();
        System.out.println("Maintenance: " + tw.getMaintenance());
        System.out.println("Average: " + tw.getAverage());

        FourWheeler fw = new FourWheeler();
        fw.setInfo(9000, 876646, "Silver");
        fw.getConsumption(10);
        fw.setSpecs(20, 5000);
        System.out.println("For Four Wheelers:");
        fw.displayInfo();
        fw.displayConsumption();
        System.out.println("Maintenance: " + fw.getMaintenance());
        System.out.println("Average: " + fw.getAverage());
    }
}
```



```
guest@user-user:~$ javac Vehiclemain.java
guest@user-user:~$ java Vehiclemain
for two wheelers
Vehicle 1490
Insurance no: 123432
color Blue
fuel consumption:5.0
maintenance: 1200.0
average 55.0
for four wheelers
Vehicle 9000
Insurance no: 876646
color Silver
fuel consumption:10.0
maintenance: 5000.0
average 20.0
```

6. Extend the above Two Wheeler class with methods getType() and getName() which gives the information about the type and the name of the company.Crea sub-classes Geared and Non Geared with method average() to print the average of a geared and non-geared two wheeler.

```
class Vehicle {
    int vno;
    int ino;
    String color;
    double fuel;

    void setInfo(int vno, int ino, String color) {
        this.vno = vno;
        this.ino = ino;
        this.color = color;
    }

    void getConsumption(double fuel) {
        this.fuel = fuel;
    }

    void displayConsumption() {
        System.out.println("Fuel consumption: " + fuel);
    }

    void displayInfo() {
        System.out.println("Vehicle number: " + vno);
        System.out.println("Insurance number: " + ino);
        System.out.println("Color: " + color);
    }
}

class TwoWheeler extends Vehicle {
    double avg;
    double mt;

    void setSpecs(double avg, double mt) {
        this.avg = avg;
        this.mt = mt;
    }

    double getMaintenance() {
        return mt;
    }

    double getAverage() {
        return avg;
    }
}
class Geared extends TwoWheeler {
    String type;
    String name;

    Geared(String type, String name) {
        this.type = type;
        this.name = name;
    }

    String getType() {
        return type;
    }

    String getName() {
        return name;
    }
}

class NonGeared extends TwoWheeler {
    String type;
    String name;

    NonGeared(String type, String name) {
        this.type = type;
        this.name = name;
    }

    String getType() {
        return type;
    }

    String getName() {
        return name;
    }
}

class VehicleMain {
    public static void main(String[] args) {
        TwoWheeler tw = new TwoWheeler();
        tw.setInfo(1490, 123432, "Blue");
        tw.getConsumption(5);
        tw.setSpecs(55, 1200);
        System.out.println("For Two Wheelers:");
        tw.displayInfo();
        tw.displayConsumption();
        System.out.println("Maintenance: " + tw.getMaintenance());
        System.out.println("Average: " + tw.getAverage());
          Geared g = new Geared("Geared", "Motorcycle");
          System.out.println("\nFor Geared Vehicle:");
          System.out.println("Type: " + g.getType());
          System.out.println("Name: " + g.getName());

          NonGeared ng = new NonGeared("Non-Geared", "Scooter");
          System.out.println("\nFor Non-Geared Vehicle:");
          System.out.println("Type: " + ng.getType());
          System.out.println("Name: " + ng.getName());
    }
}
```

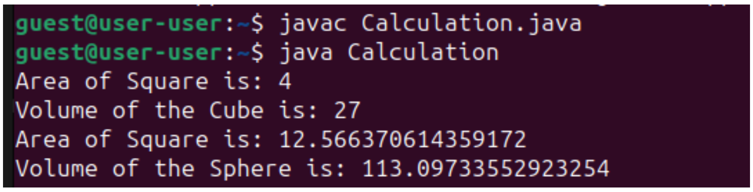


```
guest@user-user:~$ javac VehicleMain.java
guest@user-user:~$ java VehicleMain
for two wheelers
Vehicle 1490
Insurance no: 123432
color Blue
fuel consumption:5.0
maintenance: 1200.0
average 55.0

 type:geared
 name:motorcycle

 type:non-geared
 name:motorcycle
```

Week-VII

1. Create an abstract class Shape which calculates the area and volume of 2-d and 3-d shapes with methods getArea() and getVolume(). Reuse this class to calculate the area and volume of square ,circle ,cube and sphere.

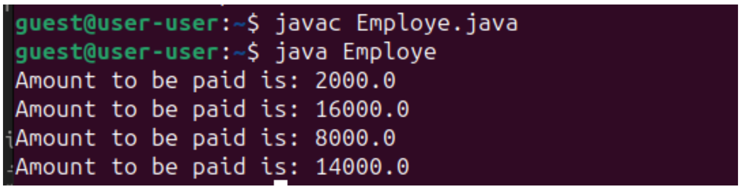
```
abstract class Shape{
  abstract void getArea(int length);
  abstract void getVolume(int length);
}
abstract class Square extends Shape{
  public void getArea(int length){
    System.out.println("Area of Square is: "+(length*length));
  }
}
abstract class Circle extends Shape{
  public void getArea(int radius){
    System.out.println("Area of Square is: "+(Math.PI*radius*radius));
  }
}
class Cube extends Square{
  public void getVolume(int side){
    System.out.println("Volume of the Cube is: "+(side*side*side));
  }
}
class Sphere extends Circle{
  public void getVolume(int radius){
    System.out.println("Volume of the Sphere is: "+((4/3.0)*Math.PI*(radius*radius*radius)) );
  }
}
class Calculation{
  public static void main(String...arr){
    Cube obj1 = new Cube();
    Sphere obj2 = new Sphere();
    obj1.getArea(2);
    obj1.getVolume(3);
    obj2.getArea(2);
    obj2.getVolume(3);
  }
}
```



```
guest@user-user:~$ javac Calculation.java
guest@user-user:~$ java Calculation
Area of Square is: 4
Volume of the Cube is: 27
Area of Square is: 12.566370614359172
Volume of the Sphere is: 113.09733552923254
```

2. Create an abstract class Employee with methods getAmount() which displays the amount paid to employees. Reuse this class to calculate the amount to be paid to WeeklyEmployee and HourlyEmployee according to no. of hours for HourlyEmployee and no. of weeks for WeeklyEmployee.

```
abstract class Employee{
  abstract void getAmount();
}
class HourlyEmployee extends Employee{
  int hours;
  float amountPerHour;
  HourlyEmployee(int hours, float amount){
    this.hours = hours;
    amountPerHour = amount;
  }
  public void getAmount(){
    System.out.println("Amount to be paid is: "+(hours*amountPerHour));
  }
}
class WeeklyEmployee extends Employee{
  int weeks;
  float amountPerWeek;
  WeeklyEmployee(int weeks, float amount){
    this.weeks = weeks;
    amountPerWeek = amount;
  }
  public void getAmount(){
    System.out.println("Amount to be paid is: "+(weeks*amountPerWeek));
  }
}
class Lab2{
  public static void main(String...arr){
    HourlyEmployee obj1 = new HourlyEmployee(20,100);
    WeeklyEmployee obj2 = new WeeklyEmployee(4,4000);
    obj1.getAmount();
    obj2.getAmount();
    obj1 = new HourlyEmployee(40,200);
```

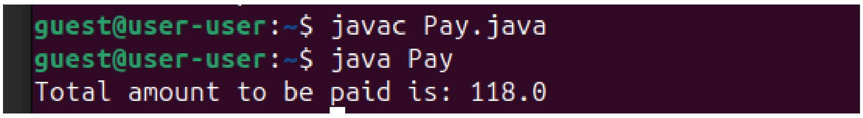


```
guest@user-user:~$ javac Employe.java
guest@user-user:~$ java Employe
Amount to be paid is: 2000.0
Amount to be paid is: 16000.0
Amount to be paid is: 8000.0
Amount to be paid is: 14000.0
```

```
    obj2 = new WeeklyEmployee(4,3500);
    obj1.getAmount();
    obj2.getAmount();
  }
}
```

3. Create an Interface payable with method getAmount ().Calculate the amount to be paid to Invoice and Employee by implementing Interface.

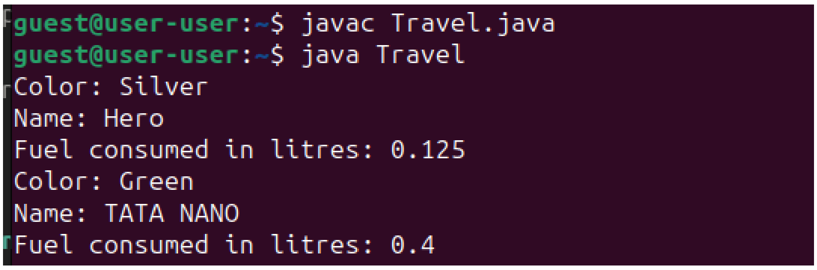
```
interface Payable{
  void getAmount(int val);
}

class Employee implements Payable{
  float totalAmount;
  public void getAmount(int amount){
    totalAmount = (float)amount+(.18f*amount);
    System.out.println("Total amount to be paid is: "+totalAmount);
  }
}

class Pay{
  public static void main(String...arr){
    Employee obj = new Employee();
    obj.getAmount(100);
  }
}
```



```
guest@user-user:~$ javac Pay.java
guest@user-user:~$ java Pay
Total amount to be paid is: 118.0
```

4. Create an Interface Vehicle with methods getColor(),getNumber(), getConsumption() calculate the fuel consumed, name and color for TwoWheeler and Four Wheeler By implementing interface Vehicle.

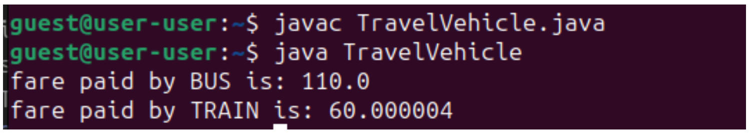
```
interface Vehicle{
  void getColor();
  void getName();
  void getConsumption(int km);
}

class TwoWheeler implements Vehicle{
  String color="Silver", name="Hero";
  double fuel;
  public void getColor(){
    System.out.println("Color: "+color);
  }
  public void getName(){
    System.out.println("Name: "+name);
  }
  public void getConsumption(int km){
    fuel = (km/40.0);
    System.out.println("Fuel consumed in litres: "+fuel);
  }
}

class FourWheeler implements Vehicle{
  String color="Green", name="TATA NANO";
  double fuel;
  public void getColor(){
    System.out.println("Color: "+color);
  }
  public void getName(){
    System.out.println("Name: "+name);
  }
  public void getConsumption(int km){
    fuel = (km/25.0);
    System.out.println("Fuel consumed in litres: "+fuel);
  }
}

class Travel{
  public static void main(String...arr){
    TwoWheeler obj1 = new TwoWheeler();
    obj1.getColor();
    obj1.getName();
    obj1.getConsumption(5);
    FourWheeler obj2 = new FourWheeler();


    obj2.getColor();
    obj2.getName();
    obj2.getConsumption(10);
  }
}
```



```
guest@user-user:~$ javac Travel.java
guest@user-user:~$ java Travel
Color: Silver
Name: Hero
Fuel consumed in litres: 0.125
Color: Green
Name: TATA NANO
Fuel consumed in litres: 0.4
```

5. Create an Interface Fare with method getAmount() to get the amount paid for fare of traveling. Calculate the fare paid by bus and train implementing interface Fare.

```
interface Fare {
    void getAmount(int km);
}

class Bus implements Fare {
    float totalAmount;

    @Override
    public void getAmount(int distance) {
        totalAmount = distance * 1.1f;
        System.out.println("Fare paid by BUS is: " + totalAmount);
    }
}

class Train implements Fare {
    float totalAmount;

    @Override
    public void getAmount(int distance) {
        totalAmount = distance * 0.6f;
        System.out.println("Fare paid by TRAIN is: " + totalAmount);
    }
}

public class TravelVehicle {
    public static void main(String[] args) {
        Bus bus = new Bus();
        Train train = new Train();

        int distance = 100; // Distance in kilometers

        bus.getAmount(distance);
        train.getAmount(distance);
    }
}
```

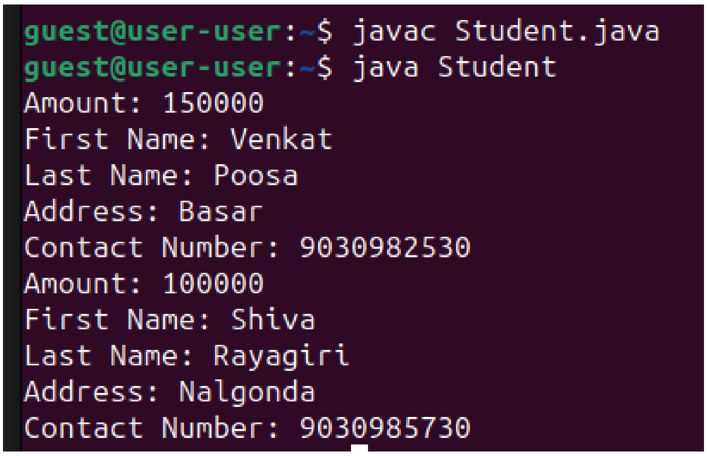


```
guest@user-user:~$ javac TravelVehicle.java
guest@user-user:~$ java TravelVehicle
fare paid by BUS is: 110.0
fare paid by TRAIN is: 60.000004
```

6. Create an Interface StudentFee with methods getAmount(), getFirstName(), getLastName() , getAddress(), getContact(). Calculate the amount paid by the Hostler and Non Hostler student by implementing interface Student Fee

```
interface StudentFee {
    void getAmount();
    void getFirstName();
    void getLastName();
    void getAddress();
    void getContact();
}

class Hostler implements StudentFee {
    String firstName, lastName, address;
    int amount;
    long contact;

    Hostler(String fName, String lName, String address, long number) {
        firstName = fName;
        lastName = lName;
        this.address = address;
        contact = number;
        amount = 150000;
    }

    public void getAmount() {
        System.out.println("Hostler Amount: " + amount);
    }

    public void getFirstName() {
        System.out.println("First Name: " + firstName);
    }

    public void getLastName() {
        System.out.println("Last Name: " + lastName);
    }

    public void getAddress() {
        System.out.println("Address: " + address);
    }

    public void getContact() {
        System.out.println("Contact Number: " + contact);
    }
}
class NonHostler implements StudentFee {
    String firstName, lastName, address;
    int amount;
    long contact;

    NonHostler(String fName, String lName, String address, long number) {
        firstName = fName;
        lastName = lName;
        this.address = address;
        contact = number;
        amount = 100000;
    }

    public void getAmount() {
        System.out.println("Non-Hostler Amount: " + amount);
    }

    public void getFirstName() {
        System.out.println("First Name: " + firstName);
    }

    public void getLastName() {
        System.out.println("Last Name: " + lastName);
    }

    public void getAddress() {
        System.out.println("Address: " + address);
    }

    public void getContact() {
        System.out.println("Contact Number: " + contact);
    } }
```

```
public class Student{
    public static void main(String... arr) {
        Hostler hostler = new Hostler("Sanjay", "Amaravadi", "Basar", 9030982530L);
        NonHostler nonHostler = new NonHostler("Rajesh", "Katnam", "Nalgonda", 9030985730L);

        System.out.println("Details of Hostler:");
        hostler.getAmount();
        hostler.getFirstName();
        hostler.getLastName();
        hostler.getAddress();
        hostler.getContact();
        System.out.println("\nDetails of Non-Hostler:");
        nonHostler.getAmount();
        nonHostler.getFirstName();
        nonHostler.getLastName();
        nonHostler.getAddress();
        nonHostler.getContact();
    }
}
```



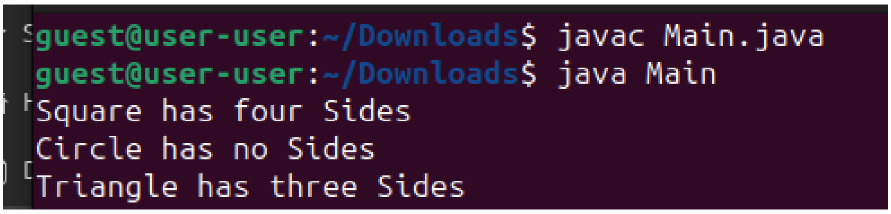
```
guest@user-user:~$ javac Student.java
guest@user-user:~$ java Student
Amount: 150000
First Name: Venkat
Last Name: Poosa
Address: Basar
Contact Number: 9030982530
Amount: 100000
First Name: Shiva
Last Name: Rayagiri
Address: Nalgonda
Contact Number: 9030985730
```

Week-VIII

1. Write a Program to create your own package. Package should have more than two classes.write a Program that uses the classes from the package.

2. Create a package named org.shapes. Create some classes in the package representing some common geometric shapes like Square, Triangle, Circle and so on. write a Program that uses the classes from the package.

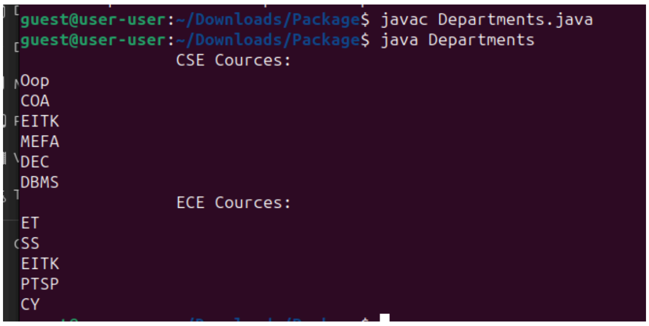
```
package org.shapes;
public class Triangle{
 public void noOfSides(){
  System.out.println("Triangle has three Sides");
 }
}
package org.shapes;
public class Circle{
 public void noOfSides(){
  System.out.println("Circle has no Sides");
 }
}
package org.shapes;
public class Square{
 public void noOfSides(){
  System.out.println("Square has four Sides");
 }
}

import org.shapes.*;
public class Main{
  public static void main(String[] args){
    Square A = new Square();
    A.noOfSides();
    Circle B = new Circle();
    B.noOfSides();
    Triangle C=new Triangle();
    C.noOfSides();
  }
}
```



```
guest@user-user:~/Downloads$ javac Main.java
guest@user-user:~/Downloads$ java Main
Square has four Sides
Circle has no Sides
Triangle has three Sides
```

3. Write a Java program to create a package called dept. Create four classes as CSE, ECE, ME and CE adds methods in each class which can display subject names of your respective year. access this package classes from main class

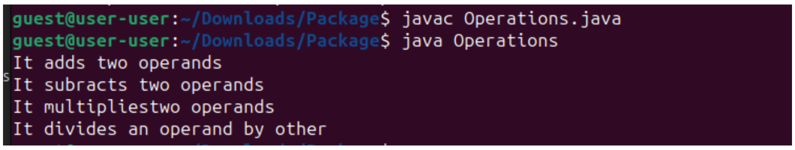
```
package dept;
public class CSE{
public void courses(){
System.out.println("       CSE Cources:");
System.out.println("Oop\nCOA\nEITK\nMEFA\nDEC\nDBMS");
}
}

package dept;
public class ME{
public void courses(){
System.out.println("       CSE Cources:");
System.out.println("KOM\nSOM\nME\nSU\nTD");
}
}

package dept;
public class ECE{
public void courses(){
System.out.println("       ECE Cources:");
System.out.println("ET\nSS\nEITK\nPTSP\nCY");
}
}

import dept.*;
public class Departments{
  public static void main(String[] args){
    CSE A = new CSE();
    A.courses();
    ECE B = new ECE();
    B.courses();
  }
}
```



```
guest@user-user:~/Downloads/Package$ javac Departments.java
guest@user-user:~/Downloads/Package$ java Departments
                CSE Cources:
Oop
COA
EITK
MEFA
DEC
DBMS
                ECE Cources:
ET
SS
EITK
PTSP
CY
```

4. Write a Calculator program : Include all calculator operations as classes in a Package "Calculator" and import it to the main class.

```
package Calculator;
public class Addition{
public void calculate(){
System.out.println("It adds two operands");
}
}

package Calculator;
public class Subraction{
public void calculate(){
System.out.println("It subracts two operands");
}
}

package Calculator;
public class Multiplication{
public void calculate(){
  System.out.println("It multipliestwo operands");
  }
  }

  package Calculator;
  public class Division{
  public void calculate(){
  System.out.println("It divides an operand by other");
  }
  }

  import Calculator.*;
  public class Operations{
    public static void main(String[] args){
      Addition A = new Addition();
      A.calculate();
      Subraction s=new Subraction();
      s.calculate();
      Multiplication m=new Multiplication();
      m.calculate();
      Division d=new Division();
      d.calculate();

    }
  }
```



```
guest@user-user:~/Downloads/Package$ javac Operations.java
guest@user-user:~/Downloads/Package$ java Operations
It adds two operands
It subracts two operands
It multipliestwo operands
It divides an operand by other
```

5. Write a program for the following

a. Example to use interfaces in Packages.

b. Example to create a sub package in a package.

a.

```
package collection;
interface Postable{
  void address(String s);
}

public class Interface implements Postable{
  public void address(String str){
    System.out.println("Letter is posted to: "+str);
  } }
```

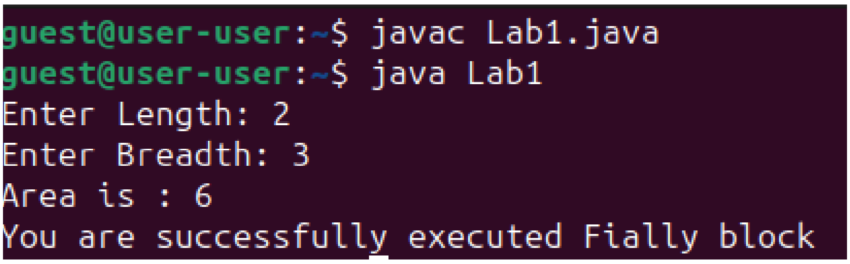
b.

```
package mypackage.subpackage;

public class SubClass {
    public void displayMessage() {
        System.out.println("This is a message from the sub-package!");
    }
}

package mypackage;

import mypackage.subpackage.SubClass;

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

Week-IX

1. Program for demonstrating the use of throw, throws & finally - Create a class with a main( ) that throws an object of class Exception inside a try block. Give the constructor for Exception a String argument. Catch the exception inside a catch clause and print the String argument. Add a finally clause and print a message to prove you were there.

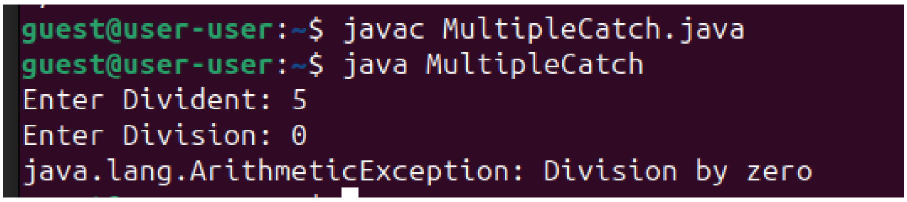
```
import java.util.*;

class MyException extends Exception {
 MyException ( String str) {
  super(str);
 }
}

class Rectangle {
 //It's mandatory to declare the checked and User-defined exceptions
 //Usage of throws is to declare the Exception
 void findArea(int length, int breadth) throws MyException {
  if(length < 0 || breadth < 0)
   //throw is used to throw an Exception intentionally (Explicitly)
   throw new MyException("Invalid Measurements");
  System.out.println("Area is : "+(length*breadth));
 }
}

class Lab1 {
 public static void main (String...arr) {
  Scanner scan = new Scanner (System.in);
  System.out.print("Enter Length: ");
  int length = scan.nextInt();
  System.out.print("Enter Breadth: ");
  int breadth = scan.nextInt();
  Rectangle obj = new Rectangle();
  //Try is used when there is a chance of getting exception
  //As we metioned, findArea throws an Exception so we keep it in try block
  try{
   //Critical statements are placed inside the try block
   obj.findArea(length,breadth);
  }
  catch (MyException e) {
   System.out.println(e);
  }
  catch (Exception e) {
   System.out.println(e.getMessage());
  }

   finally {
    System.out.println("You are successfully executed Fially block");
    scan.close();
   }
  }
 }
```



```
guest@user-user:~$ javac Lab1.java
guest@user-user:~$ java Lab1
Enter Length: 2
Enter Breadth: 3
Area is : 6
You are successfully executed Fially block
```

2. Write a program that shows that the order of the catch blocks is important. If you try to catch a superclass exception type before a subclass type, the compiler should generate errors.

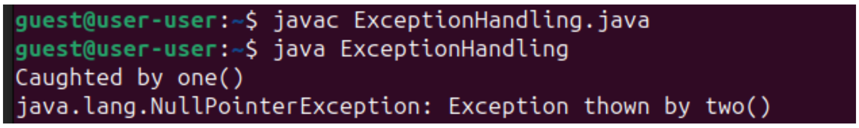
```
import java.util.*;

class Calculator {
  void findQuotient(int a, int b) {
    if(b == 0)
      throw new ArithmeticException("Division by zero");
    System.out.println(a+"/"+b+" is: "+(a/b));
  }
}
class Lab2 {
  public static void main (String...arr) {
    Scanner scan = new Scanner (System.in);
    System.out.print("Enter Divident: ");
    int num1 = scan.nextInt();
    System.out.print("Enter Division: ");
    int num2 = scan.nextInt();
    Calculator obj = new Calculator();
    try {
      obj.findQuotient(num1, num2);
    }
    //Order of Exception is important
    //Most specific exceptions are placed first
    catch(Exception e) {
      System.out.println(e);
    }
    catch(ArithmeticException e) {
      System.out.println(e);
    }
    catch(NullPointerException e) {
      System.out.println(e);
    }
    finally{
      scan.close();
    }
  }
}
```



```
guest@user-user:~$ javac MultipleCatch.java
guest@user-user:~$ java MultipleCatch
Enter Divident: 5
Enter Division: 0
java.lang.ArithmeticException: Division by zero
```

3. Write a program to rethrow an exception – Define methods one() & two(). main() should call one() and Method one() should call two(), Method two should throw an exception. Method one() should catch the exception and rethrow it to main() and main() should catch the rethrown exception.

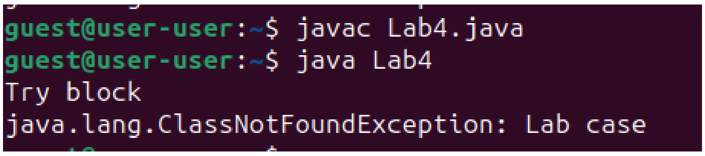
```
class Rethrow {
  void two() {
    throw new NullPointerException ("Exception thown by two()");
  }
  void one() {
    try {
      two();
    }
    catch (NullPointerException e) {
      System.out.println("Caughted by one()");
      throw e;
    }
  }
}
class Lab3 {
  public static void main (String... arr) {
    Rethrow obj = new Rethrow();
    try {
      obj.one();
    }
    catch (Exception e){
      System.out.println(e);
    }
  }
}
```



```
guest@user-user:~$ javac ExceptionHandling.java
guest@user-user:~$ java ExceptionHandling
Caughted by one()
java.lang.NullPointerException: Exception thown by two()
```

4. Exception Handling program for ClassNotFoundException--thrown if a program can not find a class it depends on at runtime (i.e., the class's ".class" file cannot be found or was removed from the CLASSPATH).

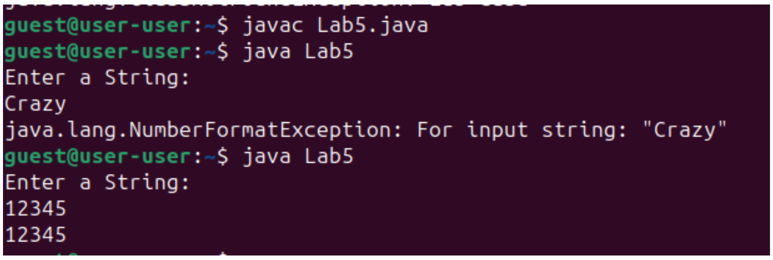
```
class Lab4 {
  public static void main (String...arr) {
    try {
      System.out.println("Try block");
      //Pass .class file name present in the current directory (or)
      //Give the path of the .class file without using double codes
      Class.forName("Lab1");
    }
    catch (ClassNotFoundException e) {
      System.out.println(e);
    }
  }
}
```



```
guest@user-user:~$ javac Lab4.java
guest@user-user:~$ java Lab4
Try block
java.lang.ClassNotFoundException: Lab case
```

5. Exception Handling program for NumberFormatException--thrown if a program is attempting to convert a string to a numerical data type, and the string contains inappropriate characters (i.e. 'z' or 'Q').

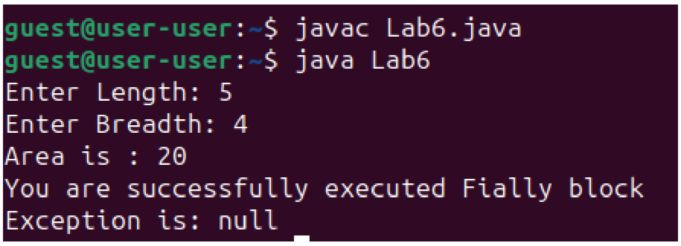
```
import java.util.*;

class Conversion {
  void toInt (String s) {
    Integer arr = Integer.parseInt(s);
    System.out.println(arr.toString());
  }
}
class Lab5 {
  public static void main (String... arr) {
    Scanner scan = new Scanner(System.in);
    System.out.println("Enter a String: ");
    String str = scan.nextLine();
    Conversion obj = new Conversion();
    try {
      obj.toInt(str);
    }
    catch (NumberFormatException e) {
      System.out.println(e);
    }
    catch (Exception e) {
      System.out.println(e);
    }
    finally{
      scan.close();
    }
  }
}
```



```
guest@user-user:~$ javac Lab5.java
guest@user-user:~$ java Lab5
Enter a String:
Crazy
java.lang.NumberFormatException: For input string: "Crazy"
guest@user-user:~$ java Lab5
Enter a String:
12345
12345
```

6. Create your own exception class using the extends keyword. Write a constructor for this class that takes a String argument and stores it inside the object with a String reference. Write a method that prints out the stored String. Create a try- catch clause to exercise your new exception.

```
import java.util.*;

class MyException extends Exception {
  static String reference;
  MyException ( String str) {
    super(str);
    reference = str;
  }
  static void display() {
    System.out.println("Exception is: "+reference);
  }
}

class Rectangle {
  //It's mandatory to declare the checked and User-defined exceptions
  //Usage of throws is to declare the Exception
  void findArea(int length, int breadth) throws MyException {
    if(length < 0 || breadth < 0)
      //throw is used to throw an Exception intentionally (Explicitly)
      throw new MyException("Invalid Measurements");
    System.out.println("Area is : "+(length*breadth));
  }
}

class Lab6 {
  public static void main (String...arr) {
    Scanner scan = new Scanner (System.in);
    System.out.print("Enter Length: ");
    int length = scan.nextInt();
    System.out.print("Enter Breadth: ");
    int breadth = scan.nextInt();
    Rectangle obj = new Rectangle();
    //Try is used when there is a chance of getting exception
    //As we metioned, findArea throws an Exception so we keep it in try block
    try{
      //Critical statements are placed inside the try block
      obj.findArea(length,breadth);
    }
    catch (MyException e) {
      System.out.println(e);
    }
    catch (Exception e) {
      System.out.println(e.getMessage());
     }
     finally {
       System.out.println("You are successfully executed Fially block");
       MyException.display();
       scan.close();
     }
    }
  }
```

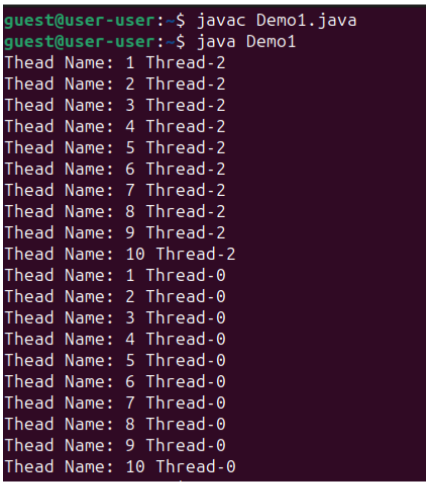


```
guest@user-user:~$ javac Lab6.java
guest@user-user:~$ java Lab6
Enter Length: 5
Enter Breadth: 4
Area is : 20
You are successfully executed Fially block
Exception is: null
```

Week-X

1. Write a program to create MyThread class with run() method and then attach a thread to this MyThread class object.

```
class MyThread extends Thread {
  public void run () {
    for(int i=0; i<10; i++)
      System.out.println ("Thead Name: "+(i+1)+" "+Thread.currentThread().getName());
  }
}
class Demo1 {
  public static void main (String ...arr) throws Exception {
    MyThread t1 = new MyThread();
    MyThread t2 = new MyThread();
    MyThread t0 = new MyThread();

    t0.start();
    t0.join();
    t1.start();

  }
}
```

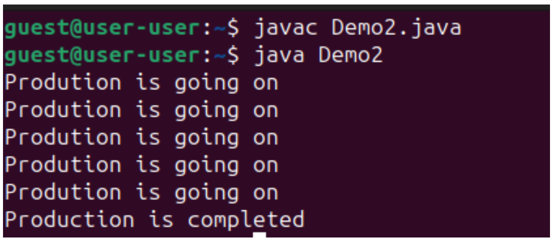


```
guest@user-user:~$ javac Demo1.java
guest@user-user:~$ java Demo1
Thead Name: 1 Thread-2
Thead Name: 2 Thread-2
Thead Name: 3 Thread-2
Thead Name: 4 Thread-2
Thead Name: 5 Thread-2
Thead Name: 6 Thread-2
Thead Name: 7 Thread-2
Thead Name: 8 Thread-2
Thead Name: 9 Thread-2
Thead Name: 10 Thread-2
Thead Name: 1 Thread-0
Thead Name: 2 Thread-0
Thead Name: 3 Thread-0
Thead Name: 4 Thread-0
Thead Name: 5 Thread-0
Thead Name: 6 Thread-0
Thead Name: 7 Thread-0
Thead Name: 8 Thread-0
Thead Name: 9 Thread-0
Thead Name: 10 Thread-0
```

2. Write a program where the consumer thread checks the data production status [is over or not] for every 10 ms.

```
 class Product {
   static Boolean status = true;
   static public void productionStatus(){
     if(status)
       System.out.println("Prodution is going on");
     else
       System.out.println("Production is completed");
   }
   static void update() {
     try { Thread.sleep(10000); }
     catch (Exception e) { }
     status = false;
   }
 }

 class MyThread extends Thread{
   public void run(){
     while(true) {
       Product.productionStatus();
       if(Product.status == false)
         break;
       try { Thread.sleep(2000); }
       catch (Exception e) { }
     }
   }
 }
 class Demo2 {
   public static void main (String...arr) throws Exception{
     //Functional Interface (Interface with only one abstract method)
     //Lambada Expression (Applicable only for Funtional Interface)
     Runnable r = () -> {
       Product.update();
```

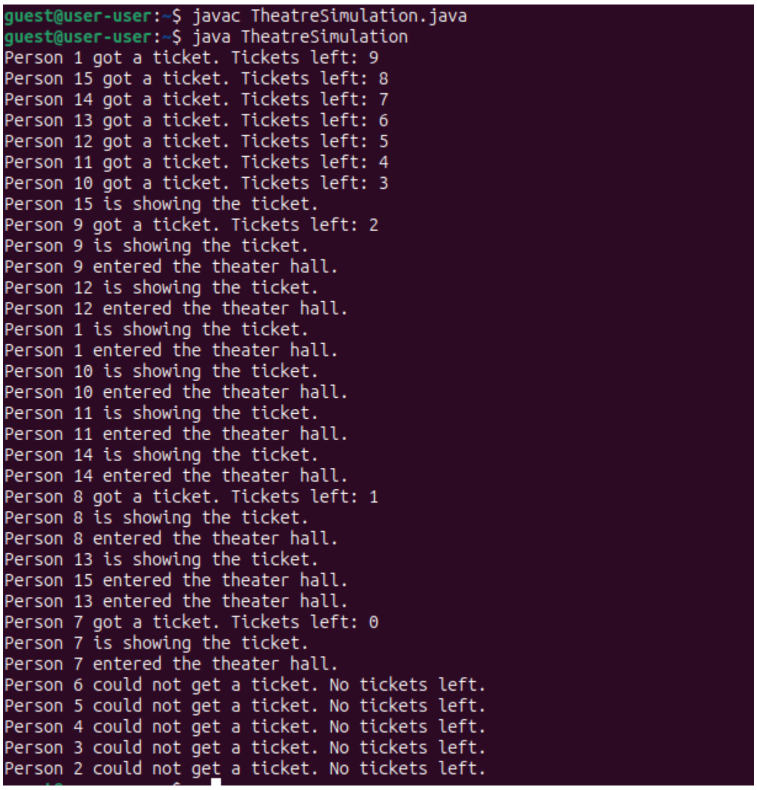
```
    };
    MyThread t1 = new MyThread();
    t1.start();
    r.run();

  }
}
```



```
guest@user-user:~$ javac Demo2.java
guest@user-user:~$ java Demo2
Prodution is going on
Prodution is going on
Prodution is going on
Prodution is going on
Prodution is going on
Production is completed
```

3. Write a Program using Threads to simulate a traffic light. The Signal lights should glow after each 10 seconds, one by one. For example: Firstly Red, then after 10 seconds, red will be put off and yellow will start glowing and then accordingly green.

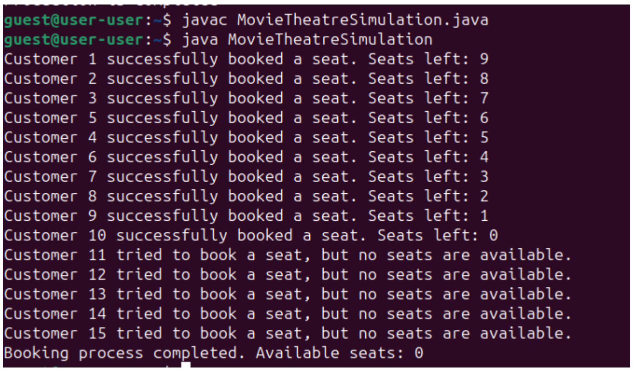
```
import java.time.*;

class TrafficLignts {
  static String color = "Red";
  static void updating() throws Exception{
    while(true) {
      color = "Red";
      Thread.sleep(10000);
      color = "Orange";
      Thread.sleep(10000);
      color = "Green";
      Thread.sleep(10000);
    }
  }

  static void status() {
    System.out.println(color);
  }
}

class Demo3 {
  public static void main (String...arr) {
    Thread t1 = new Thread(){
      public void run () {
        try { TrafficLignts.updating(); }
        catch (Exception e) { }
      }
    };
    t1.start();
    while(true) {TrafficLignts.status();
      try { Thread.sleep(2000); }
      catch (Exception e) { }
    }
  }
}
```



```
guest@user-user:~$ javac TheatreSimulation.java
guest@user-user:~$ java TheatreSimulation
Person 1 got a ticket. Tickets left: 9
Person 15 got a ticket. Tickets left: 8
Person 14 got a ticket. Tickets left: 7
Person 13 got a ticket. Tickets left: 6
Person 12 got a ticket. Tickets left: 5
Person 11 got a ticket. Tickets left: 4
Person 10 got a ticket. Tickets left: 3
Person 15 is showing the ticket.
Person 9 got a ticket. Tickets left: 2
Person 9 is showing the ticket.
Person 9 entered the theater hall.
Person 12 is showing the ticket.
Person 12 entered the theater hall.
Person 1 is showing the ticket.
Person 1 entered the theater hall.
Person 10 is showing the ticket.
Person 10 entered the theater hall.
Person 11 is showing the ticket.
Person 11 entered the theater hall.
Person 14 is showing the ticket.
Person 14 entered the theater hall.
Person 8 got a ticket. Tickets left: 1
Person 8 is showing the ticket.
Person 8 entered the theater hall.
Person 13 is showing the ticket.
Person 15 entered the theater hall.
Person 13 entered the theater hall.
Person 7 got a ticket. Tickets left: 0
Person 7 is showing the ticket.
Person 7 entered the theater hall.
Person 6 could not get a ticket. No tickets left.
Person 5 could not get a ticket. No tickets left.
Person 4 could not get a ticket. No tickets left.
Person 3 could not get a ticket. No tickets left.
Person 2 could not get a ticket. No tickets left.
```

4. Write a Program using Threads for the following case study: Movie Theatre To watch a movie

```
import java.util.concurrent.locks.ReentrantLock;

class MovieTheatre {
    private int totalSeats;
    private int availableSeats;
    private ReentrantLock lock = new ReentrantLock();

    // Constructor to initialize theatre with total seats
    public MovieTheatre(int totalSeats) {
        this.totalSeats = totalSeats;
        this.availableSeats = totalSeats;
    }

    // Method to book a seat
    public void bookSeat(String customerName) {
        lock.lock(); // Ensures thread safety
        try {
            if (availableSeats > 0) {
                System.out.println(customerName + " successfully booked a seat. Seats left: " + (availableSeats - 1));
                availableSeats--;
            } else {
                System.out.println(customerName + " tried to book a seat, but no seats are available.");
            }
        } finally {
            lock.unlock(); // Release the lock
        }
    }

    // Method to check available seats
    public int getAvailableSeats() {
        return availableSeats;
    }
}

// Customer class implementing Runnable
class Customer implements Runnable {
    private String customerName;
    private MovieTheatre theatre;

    public Customer(String customerName, MovieTheatre theatre) {
        this.customerName = customerName;
        this.theatre = theatre;
    }
     @Override
     public void run() {
        theatre.bookSeat(customerName);
     }
  }

  // Main class to simulate the movie theatre
  public class MovieTheatreSimulation {
     public static void main(String[] args) {
        // Create a Movie Theatre with 10 seats
        MovieTheatre theatre = new MovieTheatre(10);

        // Simulate 15 customers trying to book tickets
```

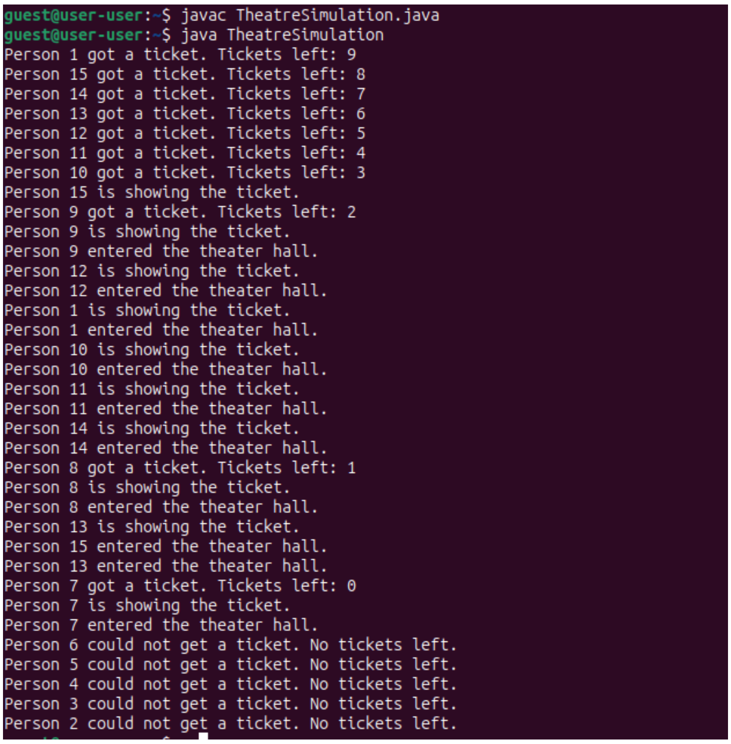


```
guest@user-user:~$ javac MovieTheatreSimulation.java
guest@user-user:~$ java MovieTheatreSimulation
Customer 1 successfully booked a seat. Seats left: 9
Customer 2 successfully booked a seat. Seats left: 8
Customer 3 successfully booked a seat. Seats left: 7
Customer 5 successfully booked a seat. Seats left: 6
Customer 4 successfully booked a seat. Seats left: 5
Customer 6 successfully booked a seat. Seats left: 4
Customer 7 successfully booked a seat. Seats left: 3
Customer 8 successfully booked a seat. Seats left: 2
Customer 9 successfully booked a seat. Seats left: 1
Customer 10 successfully booked a seat. Seats left: 0
Customer 11 tried to book a seat, but no seats are available.
Customer 12 tried to book a seat, but no seats are available.
Customer 13 tried to book a seat, but no seats are available.
Customer 14 tried to book a seat, but no seats are available.
Customer 15 tried to book a seat, but no seats are available.
Booking process completed. Available seats: 0
```

```
        Thread[] customers = new Thread[15];
        for (int i = 0; i < 15; i++) {
            customers[i] = new Thread(new Customer("Customer " + (i + 1), theatre));
        }

        // Start all customer threads
        for (Thread customer : customers) {
            customer.start();
        }

        // Wait for all threads to complete
        for (Thread customer : customers) {
            try {
                customer.join();
            } catch (InterruptedException e) {
                System.out.println("Thread interrupted: " + e.getMessage());
            }
        }

        System.out.println("Booking process completed. Available seats: " + theatre.getAvailableSeats());
    }
}
```

5. the following process is to be followed, at first get the ticket then show the ticket. Assume that N persons are trying to enter the Theatre hall all at once, displaying their sequence of entry into the theater. Note: The person should enter only after getting a ticket and showing it to the boy.

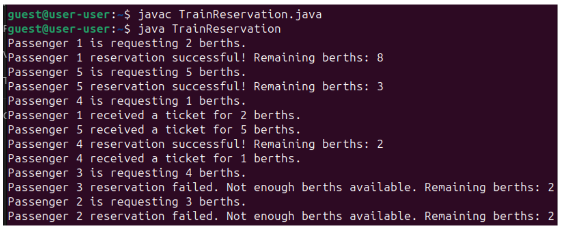
```
class Theatre {
    private int ticketsAvailable;

    public Theatre(int tickets) {
        this.ticketsAvailable = tickets;
    }

    public synchronized boolean getTicket(String person) {
        if (ticketsAvailable > 0) {
            ticketsAvailable--;
            System.out.println(person + " got a ticket. Tickets left: " + ticketsAvailable);
            return true;
        } else {
            System.out.println(person + " could not get a ticket. No tickets left.");
            return false;
        }
    }

    public void showTicket(String person) {
        System.out.println(person + " is showing the ticket.");
    }

    public void enterHall(String person) {
        System.out.println(person + " entered the theater hall.");
    }
}

class Person extends Thread {
    private String name;
    private Theatre theatre;
    }

    @Override
    public void run() {
        if (theatre.getTicket(name)) {
            theatre.showTicket(name);
            theatre.enterHall(name);
        }
    }
}
  public class TheatreSimulation {
      public static void main(String[] args) {
          int totalTickets = 10;
          int totalPersons = 15;
          Theatre theatre = new Theatre(totalTickets);

          for (int i = 1; i <= totalPersons; i++) {
              new Person("Person " + i, theatre).start();
          }
      }
  }
```



```
guest@user-user:~$ javac TheatreSimulation.java
guest@user-user:~$ java TheatreSimulation
Person 1 got a ticket. Tickets left: 9
Person 15 got a ticket. Tickets left: 8
Person 14 got a ticket. Tickets left: 7
Person 13 got a ticket. Tickets left: 6
Person 12 got a ticket. Tickets left: 5
Person 11 got a ticket. Tickets left: 4
Person 10 got a ticket. Tickets left: 3
Person 15 is showing the ticket.
Person 9 got a ticket. Tickets left: 2
Person 9 is showing the ticket.
Person 9 entered the theater hall.
Person 12 is showing the ticket.
Person 12 entered the theater hall.
Person 1 is showing the ticket.
Person 1 entered the theater hall.
Person 10 is showing the ticket.
Person 10 entered the theater hall.
Person 11 is showing the ticket.
Person 11 entered the theater hall.
Person 14 is showing the ticket.
Person 14 entered the theater hall.
Person 8 got a ticket. Tickets left: 1
Person 8 is showing the ticket.
Person 8 entered the theater hall.
Person 13 is showing the ticket.
Person 15 entered the theater hall.
Person 13 entered the theater hall.
Person 7 got a ticket. Tickets left: 0
Person 7 is showing the ticket.
Person 7 entered the theater hall.
Person 6 could not get a ticket. No tickets left.
Person 5 could not get a ticket. No tickets left.
Person 4 could not get a ticket. No tickets left.
Person 3 could not get a ticket. No tickets left.
Person 2 could not get a ticket. No tickets left.
```

6. Write a Program using Threads for the following case study: Train Reservation system To reserve a berth the following process need to be followed, at first check the number of available berths with the requested berths, if the number of requested berths are less than or equal to available berths then allot berth and print ticket or else display no berths are available.Assume that N persons are trying to reserve the berth, display their sequence of reservation status along with the number of available berths. Note : The person can print a ticket only if the berth is confirmed.

```
class TrainReservationSystem {
    private int availableBerths;

    public TrainReservationSystem(int totalBerths) {
        this.availableBerths = totalBerths;
    }

    public synchronized boolean reserveBerth(String person, int requestedBerths) {
        System.out.println(person + " is requesting " + requestedBerths + " berths.");
        if (requestedBerths <= availableBerths) {
            availableBerths -= requestedBerths;
            System.out.println(person + " reservation successful! Remaining berths: " + availableBerths);
            return true;
        } else {
                System.out.println(person + " reservation failed. Not enough berths available. Remaining berths: " +
availableBerths);
            return false;
        }
    }

    public void printTicket(String person, int requestedBerths) {
        System.out.println(person + " received a ticket for " + requestedBerths + " berths.");
    }
}

class Passenger extends Thread {
    private String name;
    private int requestedBerths;
    private TrainReservationSystem reservationSystem;
    public Passenger(String name, int requestedBerths, TrainReservationSystem reservationSystem) {
        this.name = name;
        this.requestedBerths = requestedBerths;
        this.reservationSystem = reservationSystem;
    }

    @Override
    public void run() {
        if (reservationSystem.reserveBerth(name, requestedBerths)) {
            reservationSystem.printTicket(name, requestedBerths);
        }
    }
}

public class TrainReservationSimulation {
    public static void main(String[] args) {
        int totalBerths = 10;
        TrainReservationSystem reservationSystem = new TrainReservationSystem(totalBerths);

        Passenger[] passengers = {
            new Passenger("Passenger 1", 2, reservationSystem),
            new Passenger("Passenger 2", 3, reservationSystem),
            new Passenger("Passenger 3", 4, reservationSystem),
            new Passenger("Passenger 4", 1, reservationSystem),
            new Passenger("Passenger 5", 5, reservationSystem)
        };

        for (Passenger passenger : passengers) {
            passenger.start();
        }
    }
}
```

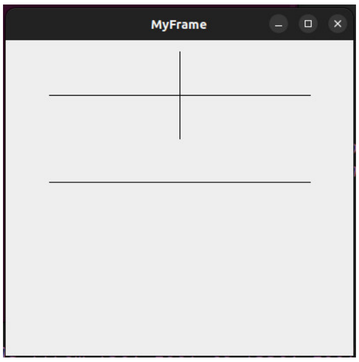


```
guest@user-user:~$ javac TrainReservation.java
guest@user-user:~$ java TrainReservation
Passenger 1 is requesting 2 berths.
Passenger 1 reservation successful! Remaining berths: 8
Passenger 5 is requesting 5 berths.
Passenger 5 reservation successful! Remaining berths: 3
Passenger 4 is requesting 1 berths.
Passenger 1 received a ticket for 2 berths.
Passenger 5 received a ticket for 5 berths.
Passenger 4 reservation successful! Remaining berths: 2
Passenger 4 received a ticket for 1 berths.
Passenger 3 is requesting 4 berths.
Passenger 3 reservation failed. Not enough berths available. Remaining berths: 2
Passenger 2 is requesting 3 berths.
Passenger 2 reservation failed. Not enough berths available. Remaining berths: 2
```

Week-XI

1. Write a program for the following.
a. display a frame with title MyFrame
b. draw a horizontal line.
c. Draw one line perpendicular to the other and One line parallel to the other.

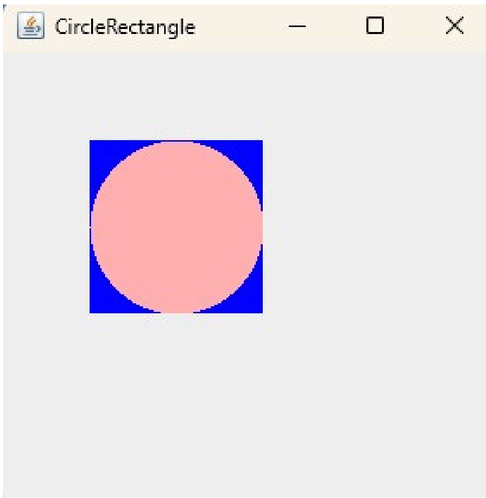
```
import javax.swing.*;
import java.awt.*;

public class MyFrame extends JFrame {

    // Constructor to setup the frame
    public MyFrame() {
        setTitle("MyFrame");  // Set the title of the frame
        setSize(400, 400);    // Set the size of the frame
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  // Close on exit
        setLocationRelativeTo(null);  // Center the frame on the screen
    }

    // Method to paint the frame content
    @Override
    public void paint(Graphics g) {
        super.paint(g);  // Call the parent class paint method

        // Draw a horizontal line (from (50, 100) to (350, 100))
        g.drawLine(50, 100, 350, 100);

        // Draw one line perpendicular to the above horizontal line (vertical line)
        g.drawLine(200, 50, 200, 150);  // (200, 50) to (200, 150)

        // Draw one line parallel to the above horizontal line
        g.drawLine(50, 200, 350, 200);  // (50, 200) to (350, 200)
    }

    public static void main(String[] args) {
        // Create an instance of MyFrame and make it visible
        SwingUtilities.invokeLater(() -> {
            MyFrame frame = new MyFrame();
            frame.setVisible(true);
        });
    }
}
```



2. Create an application to display a circle within rectangle and fill different colors in the circle & rectangle

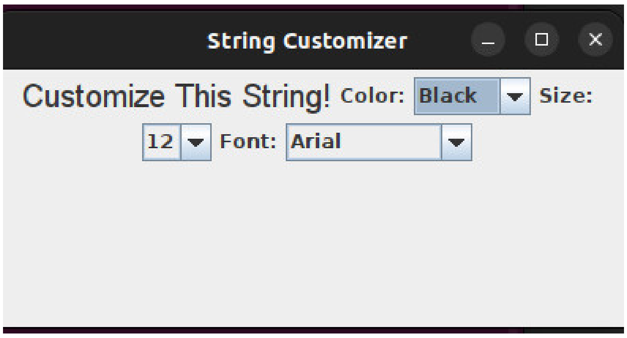
```
package week11;
import java.awt.*;
import javax.swing.*;
public class CircleRectangle extends JFrame {
 public CircleRectangle() {
  super("CircleRectangle");
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  setSize(300,300);
  add(new CirclePanel());
  setVisible(true);
  }
public static void main(String[] args) {
 CircleRectangle c=new CircleRectangle();
```

```
  }
  class CirclePanel extends JPanel{

   public void paintComponent(Graphics g) {
    super.paintComponent(g);
   Graphics2D g2d=(Graphics2D) g;
   //rectangle
   g2d.setColor(Color.BLUE);
    g2d.fillRect(50, 50, 100, 100);
    g2d.setColor(Color.PINK);
    g.fillOval(50, 50, 100, 100);
    setVisible(true);
  }
  }
  }
```



3. Write an application that displays any string. Choose color from combo box to change the color of this displayed string and choose its size & type respectively from another two combo boxes.

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

public class StringCustomizerApp {
    public static void main(String[] args) {
        // Create the frame
        JFrame frame = new JFrame("String Customizer");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new FlowLayout());

        // Create label to display the string
        JLabel label = new JLabel("Customize This String!");
        label.setFont(new Font("Arial", Font.PLAIN, 20));
        frame.add(label);

        // Create combo boxes for color, font size, and font
            JComboBox<String> colorComboBox = new JComboBox<>(new String[]{"Black", "Red", "Blue", "Green",
"Yellow"});
        JComboBox<String> sizeComboBox = new JComboBox<>(new String[]{"12", "16", "20", "24", "28"});
            JComboBox<String> fontComboBox = new JComboBox<>(new String[]{"Arial", "Serif", "Monospaced",
"Dialog"});

        // ActionListener for updating label based on selections
        ActionListener updateLabel = new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                // Get selected values from the combo boxes
                String color = (String) colorComboBox.getSelectedItem();
                int size = Integer.parseInt((String) sizeComboBox.getSelectedItem());
                String font = (String) fontComboBox.getSelectedItem();

                // Set font and size
                label.setFont(new Font(font, Font.PLAIN, size));
                switch (color) {
                    case "Red":
                        label.setForeground(Color.RED);
                        break;
                    case "Blue":
                        label.setForeground(Color.BLUE);
                        break;
                    case "Green":
                        label.setForeground(Color.GREEN);
                        break;
                    case "Yellow":
                        label.setForeground(Color.YELLOW);
                        break;
                    default:
                        label.setForeground(Color.BLACK);
                        break;
                }
            }
        };

        colorComboBox.addActionListener(updateLabel);
        sizeComboBox.addActionListener(updateLabel);
        fontComboBox.addActionListener(updateLabel);
        frame.add(new JLabel("Color:"));
        frame.add(colorComboBox);
        frame.add(new JLabel("Size:"));
        frame.add(sizeComboBox);
        frame.add(new JLabel("Font:"));
        frame.add(fontComboBox);
        frame.setSize(400, 200);
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }
}
```

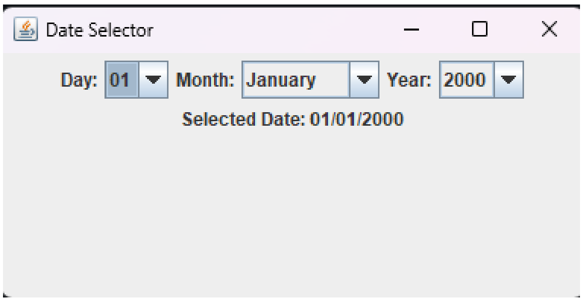


Customize This String! Color: Black Size: 12 Font: Arial

3.Write a small application with a default date 01/01/2000 and three combo boxes displaying valid days, months & year (1990 – 2050). Change the displayed date with the one chosen by user from these combo boxes.

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

class DateSelector {
    private JFrame frame;
    private JLabel dateLabel;
    private JComboBox<String> dayComboBox;
    private JComboBox<String> monthComboBox;
    private JComboBox<String> yearComboBox;

    public DateSelector() {
        initializeComponents();
    }

    private void initializeComponents() {
        frame = new JFrame("Date Selector");
        frame.setSize(400, 200);
```

```
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setLayout(new FlowLayout());

            String[] days = generateDays();
            String[] months = {
                "January", "February", "March", "April", "May", "June",
                "July", "August", "September", "October", "November", "December"
            };
            String[] years = generateYears(1990, 2050);

            dayComboBox = new JComboBox<>(days);
            monthComboBox = new JComboBox<>(months);
            yearComboBox = new JComboBox<>(years);

            dayComboBox.setSelectedIndex(0);
            monthComboBox.setSelectedIndex(0);
            yearComboBox.setSelectedItem("2000");


      dateLabel = new JLabel("Selected Date: 01/01/2000");

      ActionListener updateDateListener = e -> updateDate();
      dayComboBox.addActionListener(updateDateListener);
      monthComboBox.addActionListener(updateDateListener);
      yearComboBox.addActionListener(updateDateListener);
        frame.add(new JLabel("Day:"));
        frame.add(dayComboBox);

        frame.add(new JLabel("Month:"));
        frame.add(monthComboBox);

        frame.add(new JLabel("Year:"));
        frame.add(yearComboBox);

        frame.add(dateLabel);
    }

    private String[] generateDays() {
        String[] days = new String[31];
        for (int i = 0; i < 31; i++) {
            days[i] = String.format("%02d", i + 1);
        }
        return days;
    }

    private String[] generateYears(int startYear, int endYear) {
        String[] years = new String[endYear - startYear + 1];
        for (int i = 0; i < years.length; i++) {
            years[i] = String.valueOf(startYear + i);
        }
        return years;
    }

    private void updateDate() {
        String day = (String) dayComboBox.getSelectedItem();
        String month = (String) monthComboBox.getSelectedItem();
        String year = (String) yearComboBox.getSelectedItem();
        int monthNumber = monthComboBox.getSelectedIndex() + 1;

        dateLabel.setText(String.format("Selected Date: %02d/%02d/%s", Integer.parseInt(day), monthNumber, year));
    }

    public void show() {
        frame.setVisible(true);
    }
}

public class Main {
 public static void main(String[] args) {
 SwingUtilities.invokeLater(() -> {
 DateSelector dateSelector = new DateSelector();
 dateSelector.show();
 });
 }
}
```

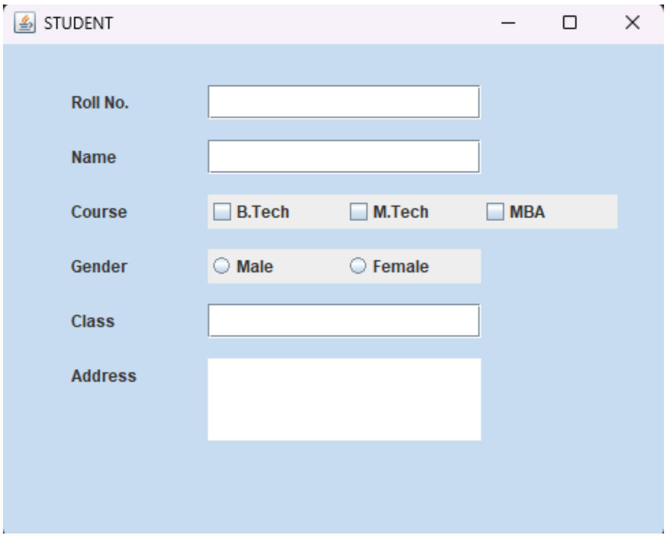


Day: 01 Month: January Year: 2000

Selected Date: 01/01/2000

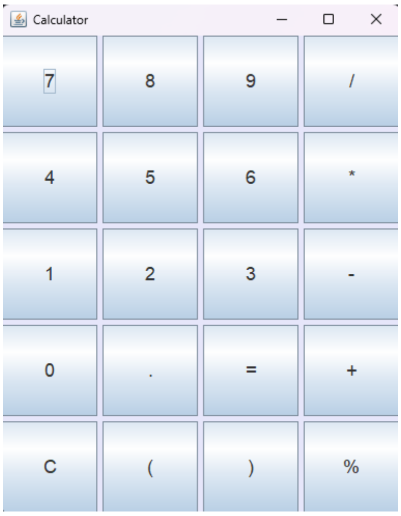
5.Create a GUI with title STUDENT which has labels roll no., name, course, gender, class, address with textboxes for taking input from the user(without any functionality) and checkboxes for selecting the course, radio buttons for selecting gender with appropriate background color.

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

public class StudentForm {
    public static void main(String[] args) {
        // Create the main frame
        JFrame frame = new JFrame("STUDENT");
        frame.setSize(500, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(null);

        // Set background color
        frame.getContentPane().setBackground(new Color(200, 220, 240));

        // Roll No. Label and TextField
        JLabel rollNoLabel = new JLabel("Roll No.");
        rollNoLabel.setBounds(50, 30, 100, 25);
        frame.add(rollNoLabel);

        JTextField rollNoField = new JTextField();
        rollNoField.setBounds(150, 30, 200, 25);
        frame.add(rollNoField);

        // Name Label and TextField
        JLabel nameLabel = new JLabel("Name");
        nameLabel.setBounds(50, 70, 100, 25);
        frame.add(nameLabel);

        JTextField nameField = new JTextField();
        nameField.setBounds(150, 70, 200, 25);
        frame.add(nameField);

        // Course Label and Checkboxes
        JLabel courseLabel = new JLabel("Course");
        courseLabel.setBounds(50, 110, 100, 25);
        frame.add(courseLabel);

        JCheckBox course1 = new JCheckBox("B.Tech");
        course1.setBounds(150, 110, 100, 25);
        frame.add(course1);

        JCheckBox course2 = new JCheckBox("M.Tech");
        course2.setBounds(250, 110, 100, 25);
        frame.add(course2);

        JCheckBox course3 = new JCheckBox("MBA");
        course3.setBounds(350, 110, 100, 25);
        frame.add(course3);

        // Gender Label and Radio Buttons
        JLabel genderLabel = new JLabel("Gender");
        genderLabel.setBounds(50, 150, 100, 25);
        frame.add(genderLabel);

        JRadioButton maleButton = new JRadioButton("Male");
        maleButton.setBounds(150, 150, 100, 25);
        frame.add(maleButton);

        JRadioButton femaleButton = new JRadioButton("Female");
        femaleButton.setBounds(250, 150, 100, 25);
        frame.add(femaleButton);

        ButtonGroup genderGroup = new ButtonGroup();
        genderGroup.add(maleButton);
        genderGroup.add(femaleButton);

        // Class Label and TextField
        JLabel classLabel = new JLabel("Class");
        classLabel.setBounds(50, 190, 100, 25);
        frame.add(classLabel);

        JTextField classField = new JTextField();
        classField.setBounds(150, 190, 200, 25);
        frame.add(classField);

        // Address Label and TextArea
        JLabel addressLabel = new JLabel("Address");
        addressLabel.setBounds(50, 230, 100, 25);
        frame.add(addressLabel);

        JTextArea addressArea = new JTextArea();
        addressArea.setBounds(150, 230, 200, 60);
        frame.add(addressArea);

        // Display the frame
        frame.setVisible(true);
    }
}
```



6.Create a GUI application to display a calculator using grid Layout (You do not have to provide functionality).

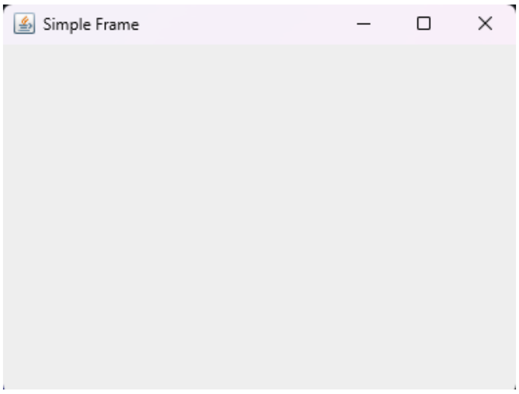
```
import javax.swing.*;
import java.awt.*;

public class CalculatorGUI {
   public static void main(String[] args) {
      // Create the main frame
      JFrame frame = new JFrame("Calculator");
      frame.setSize(400, 500);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      // Set GridLayout for the calculator
      frame.setLayout(new GridLayout(5, 4, 5, 5)); // 5 rows, 4 columns, 5px gaps

      // Add buttons for the calculator
      String[] buttons = {
         "7", "8", "9", "/",
         "4", "5", "6", "*",
         "1", "2", "3", "-",
         "0", ".", "=", "+",
         "C", "(", ")", "%"
      };

      for (String text : buttons) {
         JButton button = new JButton(text);
         button.setFont(new Font("Arial", Font.PLAIN, 18));
         frame.add(button);
      }

      // Set background color
      frame.getContentPane().setBackground(new Color(230, 230, 250)); // Light lavender

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



Week-XII

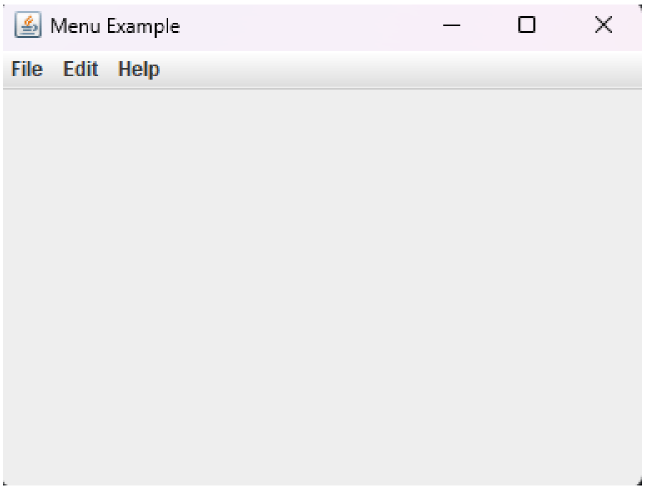
1.Write a program to create a frame by creating an object to Jframe class and include close button to terminate the application of the frame.

```
import javax.swing.*;

public class SimpleFrame {
   public static void main(String[] args) {
      // Create a JFrame object
      JFrame frame = new JFrame("Simple Frame");

      // Set the size of the frame
      frame.setSize(400, 300);

      // Set the default close operation to terminate the application
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

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



5.Write a program to create a menu with several menu items.

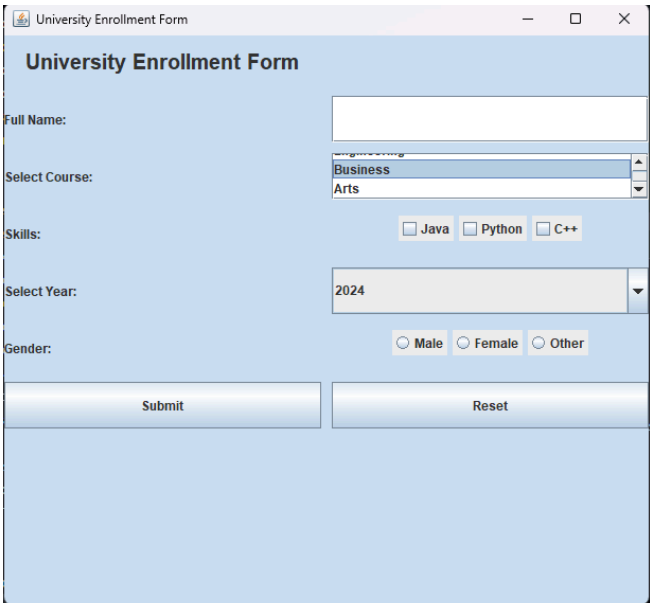
```
import javax.swing.*;
import java.awt.event.*;

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

        JFrame frame = new JFrame("Menu Example");
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JMenuBar menuBar = new JMenuBar();

        JMenu fileMenu = new JMenu("File");
        JMenu editMenu = new JMenu("Edit");
        JMenu helpMenu = new JMenu("Help");

        JMenuItem newItem = new JMenuItem("New");
        JMenuItem openItem = new JMenuItem("Open");
        JMenuItem saveItem = new JMenuItem("Save");
        JMenuItem exitItem = new JMenuItem("Exit");

        exitItem.addActionListener(e -> System.exit(0));

        fileMenu.add(newItem);
        fileMenu.add(openItem);
        fileMenu.add(saveItem);
        fileMenu.addSeparator(); // Add a separator line
        fileMenu.add(exitItem);

        JMenuItem cutItem = new JMenuItem("Cut");
        JMenuItem copyItem = new JMenuItem("Copy");
        JMenuItem pasteItem = new JMenuItem("Paste");

        editMenu.add(cutItem);
        editMenu.add(copyItem);
        editMenu.add(pasteItem);

 JMenuItem aboutItem = new JMenuItem("About");

 helpMenu.add(aboutItem);

 // Add menus to menu bar
 menuBar.add(fileMenu);
 menuBar.add(editMenu);
 menuBar.add(helpMenu);

 frame.setJMenuBar(menuBar);

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



6. Create an application Form for University Enroll ment with the following Fields. a. Check box b. Text area c. List box d. Display text e. Push buttons f. Combo box. g. Radio buttons. Back ground color

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

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

      JFrame frame = new JFrame("University Enrollment Form");
      frame.setSize(600, 600);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.getContentPane().setBackground(new Color(200, 220, 240));

      frame.setLayout(new GridLayout(10, 2, 10, 10));

      JLabel titleLabel = new JLabel("University Enrollment Form", JLabel.CENTER);
      titleLabel.setFont(new Font("Arial", Font.BOLD, 20));
      frame.add(titleLabel);
      frame.add(new JLabel("")); // Empty placeholder

      JLabel nameLabel = new JLabel("Full Name:");
      JTextArea nameArea = new JTextArea(2, 20);
      frame.add(nameLabel);
      frame.add(new JScrollPane(nameArea));

      JLabel courseLabel = new JLabel("Select Course:");
      JList<String> courseList = new JList<>(new String[]{"Computer Science", "Engineering", "Business", "Arts"});
      frame.add(courseLabel);
      frame.add(new JScrollPane(courseList));

      JLabel skillsLabel = new JLabel("Skills:");
      JPanel skillsPanel = new JPanel();
      skillsPanel.setBackground(new Color(200, 220, 240));
      JCheckBox javaCheckBox = new JCheckBox("Java");
      JCheckBox pythonCheckBox = new JCheckBox("Python");
      JCheckBox cCheckBox = new JCheckBox("C++");


      skillsPanel.add(javaCheckBox);
      skillsPanel.add(pythonCheckBox);
      skillsPanel.add(cCheckBox);
      frame.add(skillsLabel);
      frame.add(skillsPanel);

      JLabel yearLabel = new JLabel("Select Year:");
      JComboBox<String> yearComboBox = new JComboBox<>(new String[]{"2024", "2025", "2026"});
      frame.add(yearLabel);
      frame.add(yearComboBox);

      JLabel genderLabel = new JLabel("Gender:");
      JPanel genderPanel = new JPanel();
      genderPanel.setBackground(new Color(200, 220, 240));
      JRadioButton maleButton = new JRadioButton("Male");
      JRadioButton femaleButton = new JRadioButton("Female");
```

```
        JRadioButton otherButton = new JRadioButton("Other");
        ButtonGroup genderGroup = new ButtonGroup();
        genderGroup.add(maleButton);
        genderGroup.add(femaleButton);
        genderGroup.add(otherButton);
        genderPanel.add(maleButton);
        genderPanel.add(femaleButton);
        genderPanel.add(otherButton);
        frame.add(genderLabel);
        frame.add(genderPanel);

        JButton submitButton = new JButton("Submit");
        JButton resetButton = new JButton("Reset");
        frame.add(submitButton);
        frame.add(resetButton);

        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String name = nameArea.getText();
                String selectedCourse = courseList.getSelectedValue();
                String selectedYear = (String) yearComboBox.getSelectedItem();
                String gender = maleButton.isSelected() ? "Male" : femaleButton.isSelected() ? "Female" : "Other";
                StringBuilder skills = new StringBuilder();
                if (javaCheckBox.isSelected()) skills.append("Java ");
                if (pythonCheckBox.isSelected()) skills.append("Python ");
                if (cCheckBox.isSelected()) skills.append("C++");

                JOptionPane.showMessageDialog(frame, "Submission Successful!\nName: " + name +
                        "\nCourse: " + selectedCourse + "\nYear: " + selectedYear +
                        "\nGender: " + gender + "\nSkills: " + skills);
            }
        });

        resetButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                nameArea.setText("");
                courseList.clearSelection();
                yearComboBox.setSelectedIndex(0);
                genderGroup.clearSelection();
                javaCheckBox.setSelected(false);
                pythonCheckBox.setSelected(false);
                cCheckBox.setSelected(false);
            }
        });

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


University Enrollment Form

University Enrollment Form

Full Name:

Select Course: Business, Arts

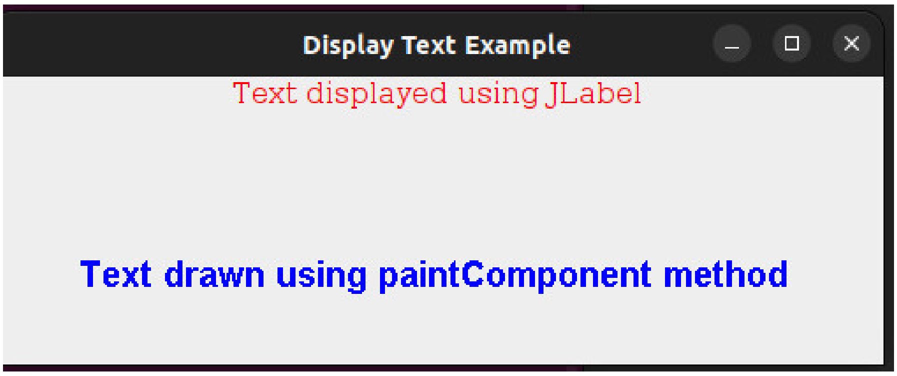
Skills: □ Java □ Python □ C++

Select Year: 2024

Gender: ○ Male ○ Female ○ Other

3. Write a program for the following.
a. Display text in the frame by overriding the PaintComponent() method of Jpanel class.
b. Display some text in the frame with the help of a Label.

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

public class DisplayTextApp {

    public static void main(String[] args) {
        // Create frame
        JFrame frame = new JFrame("Display Text Example");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new BorderLayout());

        // Panel to override paintComponent method and draw text
        JPanel panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);  // Call the parent class paintComponent
                g.setFont(new Font("Arial", Font.BOLD, 20));
                g.setColor(Color.BLUE);
                g.drawString("Text drawn using paintComponent method", 50, 100); // Display text on JPanel
            }
        };

        // Create a label to display text
        JLabel label = new JLabel("Text displayed using JLabel", SwingConstants.CENTER);
        label.setFont(new Font("Serif", Font.PLAIN, 16));
        label.setForeground(Color.RED);

        // Add the label to the frame
        frame.add(label, BorderLayout.NORTH);
        // Add the panel (with overridden paintComponent) to the frame
        frame.add(panel, BorderLayout.CENTER);

        // Set frame size and make it visible
        frame.setSize(400, 200);
        frame.setLocationRelativeTo(null);  // Center the frame on the screen
        frame.setVisible(true);
    }
}
```



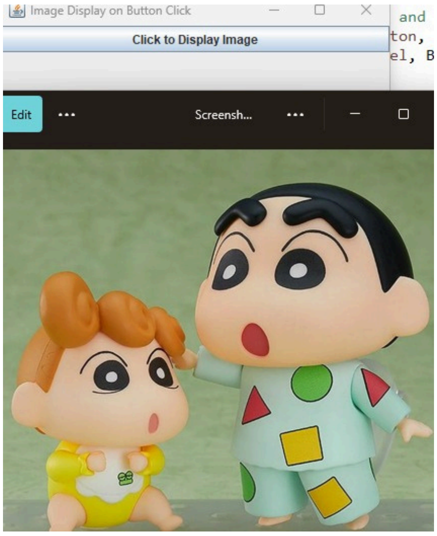
Display Text Example

Text displayed using JLabel

Text drawn using paintComponent method

4. Write a program to create a push button , when the button is clicked an image is displayed in the frame.

```
package week12;

 import javax.swing.*;
 import java.awt.*;
 import java.awt.event.ActionEvent;
 import java.awt.event.ActionListener;

 public class Mainaa {

     public static void main(String[] args) {

         JFrame frame = new JFrame("Image Display on Button Click");
         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
         frame.setSize(400, 400);
         frame.setLayout(new BorderLayout());

         JButton button = new JButton("Click to Display Image");

         JLabel label = new JLabel("", JLabel.CENTER);

         frame.add(button, BorderLayout.NORTH);
         frame.add(label, BorderLayout.CENTER);

         button.addActionListener(new ActionListener() {
             @Override
             public void actionPerformed(ActionEvent e) {

                         ImageIcon imageIcon = new ImageIcon("\"C:\\Users\\acer\\OneDrive\\Pictures\\Screenshot_31-3-
2024_123239_.jpeg\""); // Update the path
                 label.setIcon(imageIcon);
                 frame.revalidate();
             }
         });

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



Week-XIII

1. Write a program to insert data into Student Table.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Scanner;

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

        String jdbcURL = "jdbc:mysql://localhost:3306/Student";
        String dbUsername = "root";
        String dbPassword = "student";

        // SQL query to insert data into the Student table
        String insertQuery = "INSERT INTO Student (id, name, age, course) VALUES (?, ?, ?, ?);";

        // Create Scanner object for user input
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter Student ID: ");
        int id = scanner.nextInt();

        scanner.nextLine(); // Consume newline

        System.out.println("Enter Student Name: ");
        String name = scanner.nextLine();

        System.out.println("Enter Student Age: ");
        int age = scanner.nextInt();

        scanner.nextLine();

        System.out.println("Enter Student Course: ");
        String course = scanner.nextLine();

        try (Connection connection = DriverManager.getConnection(jdbcURL, dbUsername, dbPassword);
             PreparedStatement preparedStatement = connection.prepareStatement(insertQuery)) {

         // Set the values for the prepared statement
 preparedStatement.setInt(1, id);
 preparedStatement.setString(2, name);
 preparedStatement.setInt(3, age);
 preparedStatement.setString(4, course);

 // Execute the query
 int rowsInserted = preparedStatement.executeUpdate();

 if (rowsInserted > 0) {
 System.out.println("Student data inserted successfully!");
 } else {
 System.out.println("Failed to insert student data.");
 }
 } catch (SQLException e) {
  e.printStackTrace();
  System.out.println("Error connecting to the database or inserting data.");
  } finally {
  scanner.close();
  }
  }
 }
```

2.Write a program to retrieve the data from the table Student.

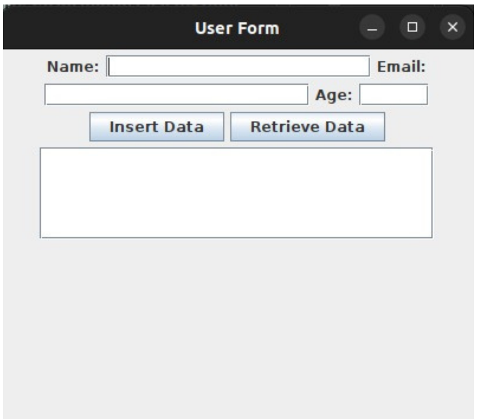
```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

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

        String jdbcURL = "jdbc:mysql://localhost:3306/Student// Replace with your database URL
        String dbUsername = "root";
        String dbPassword = "student";

        // SQL query to retrieve data from the Student table
        String selectQuery = "SELECT * FROM Student;";

        try (Connection connection = DriverManager.getConnection(jdbcURL, dbUsername, dbPassword);
             Statement statement = connection.createStatement();
             ResultSet resultSet = statement.executeQuery(selectQuery)) {

            // Display retrieved data
            System.out.println("Student Data:");
            System.out.printf("%-10s %-20s %-5s %-20s\n", "ID", "Name", "Age", "Course");

            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name = resultSet.getString("name");
                int age = resultSet.getInt("age");
                String course = resultSet.getString("course");

                System.out.printf("%-10d %-20s %-5d %-20s\n", id, name, age, course);
            }

        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("Error connecting to the database or retrieving data.");
        }
    }
}
```

3. Create a Form to insert and retrieve the data from the Database as users prefer.

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

public class UserForm extends JFrame {
```

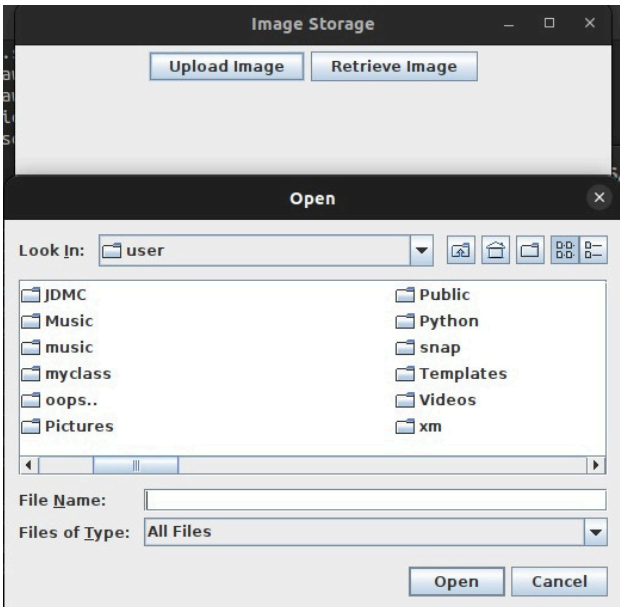
```
private JTextField nameField, emailField, ageField;
private JButton insertButton, retrieveButton;
private JTextArea resultArea;

// Database connection details
private static final String URL = "jdbc:mysql://localhost:3306/UserDataDB";
private static final String USER = "root"; // Change to your MySQL username
private static final String PASSWORD = "password"; // Change to your MySQL password

public UserForm() {
    // Setting up the frame
    setTitle("User Form");
    setSize(400, 350);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);

    // Creating the form components
    JLabel nameLabel = new JLabel("Name:");
    nameField = new JTextField(20);

    JLabel emailLabel = new JLabel("Email:");
    emailField = new JTextField(20);

    JLabel ageLabel = new JLabel("Age:");
    ageField = new JTextField(5);

    insertButton = new JButton("Insert Data");
    retrieveButton = new JButton("Retrieve Data");

    resultArea = new JTextArea(5, 30);
    resultArea.setEditable(false);

    // Layout and adding components
    setLayout(new FlowLayout());

    add(nameLabel);
    add(nameField);

    add(emailLabel);
    add(emailField);

    add(ageLabel);
    add(ageField);

    add(insertButton);
    add(retrieveButton);

    add(new JScrollPane(resultArea));

    // Event handlers for buttons
    insertButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            insertData();
        }
    });
```

```
        retrieveButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                retrieveData();
            }
        });
    }

    // Method to insert data into the database
    private void insertData() {
        String name = nameField.getText();
        String email = emailField.getText();
        String age = ageField.getText();

        if (name.isEmpty() || email.isEmpty() || age.isEmpty()) {
            JOptionPane.showMessageDialog(this, "All fields must be filled out!");
            return;
        }

        try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD)) {
            String query = "INSERT INTO Users (name, email, age) VALUES (?, ?, ?)";
            PreparedStatement pst = conn.prepareStatement(query);
            pst.setString(1, name);
            pst.setString(2, email);
            pst.setInt(3, Integer.parseInt(age));

            int rowsAffected = pst.executeUpdate();
            if (rowsAffected > 0) {
                JOptionPane.showMessageDialog(this, "Data inserted successfully!");
                clearFields();
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error inserting data: " + ex.getMessage());
        } } }
```



4. Write a program to store an Image and retrieve an image from Database

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

public class ImageStorageApp extends JFrame {

    private JButton uploadButton, retrieveButton;
    private JLabel imageLabel;
    private JFileChooser fileChooser;
    private static final String URL = "jdbc:mysql://localhost:3306/ImageDB";
    private static final String USER = "root";
    private static final String PASSWORD = "password";

    public ImageStorageApp() {
        setTitle("Image Storage");
        setSize(500, 500);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        uploadButton = new JButton("Upload Image");
        retrieveButton = new JButton("Retrieve Image");
        imageLabel = new JLabel("No Image", JLabel.CENTER);
        fileChooser = new JFileChooser();

        uploadButton.addActionListener(e -> uploadImage());
        retrieveButton.addActionListener(e -> retrieveImage());

        setLayout(new BorderLayout());
        JPanel panel = new JPanel();
        panel.add(uploadButton);
        panel.add(retrieveButton);

        add(panel, BorderLayout.NORTH);
        add(imageLabel, BorderLayout.CENTER);
        setVisible(true);
    }

    private void uploadImage() {
        int result = fileChooser.showOpenDialog(this);
        if (result == JFileChooser.APPROVE_OPTION) {
            try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
                 FileInputStream fis = new FileInputStream(fileChooser.getSelectedFile())) {
                String query = "INSERT INTO Images (name, image) VALUES (?, ?)";
                try (PreparedStatement pst = conn.prepareStatement(query)) {
                    pst.setString(1, fileChooser.getSelectedFile().getName());
                    pst.setBinaryStream(2, fis);
                    pst.executeUpdate();
                    JOptionPane.showMessageDialog(this, "Image uploaded!");
                }
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());
            }
        }
    }

    private void retrieveImage() {
        try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
             Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery("SELECT image FROM Images ORDER BY id DESC LIMIT 1")) {
            if (rs.next()) {
                byte[] imgBytes = rs.getBytes("image");
                ImageIcon icon = new ImageIcon(imgBytes);
                imageLabel.setIcon(icon);
                imageLabel.setText("");  // Clear text
            } else {
                JOptionPane.showMessageDialog(this, "No image found.");
            }
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(ImageStorageApp::new);
    }
}
```



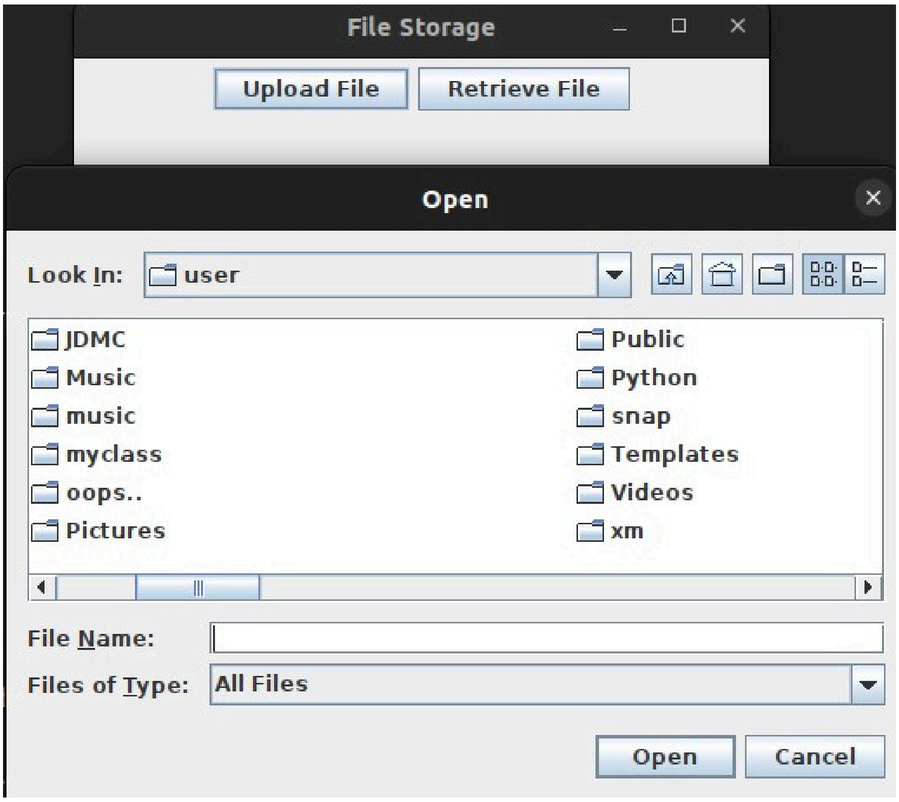
Look In: user

File Name:

Files of Type: All Files

5. Write a program to Store and retrieve file content from the Database.

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

public class FileStorageApp extends JFrame {

    private JButton uploadButton, retrieveButton;
    private JLabel statusLabel;
    private JFileChooser fileChooser;

    private static final String URL = "jdbc:mysql://localhost:3306/FileDB";
    private static final String USER = "root";
    private static final String PASSWORD = "password";

    public FileStorageApp() {
        setTitle("File Storage");
        setSize(400, 150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        uploadButton = new JButton("Upload File");
        retrieveButton = new JButton("Retrieve File");
        statusLabel = new JLabel("Select an option", JLabel.CENTER);
        fileChooser = new JFileChooser();

        uploadButton.addActionListener(e -> uploadFile());
        retrieveButton.addActionListener(e -> retrieveFile());

        JPanel panel = new JPanel();
        panel.add(uploadButton);
        panel.add(retrieveButton);

        add(panel, BorderLayout.NORTH);
        add(statusLabel, BorderLayout.CENTER);
        setVisible(true);
    }

    private void uploadFile() {
        if (fileChooser.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
            File file = fileChooser.getSelectedFile();
            try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
                 PreparedStatement pst = conn.prepareStatement("INSERT INTO Files (name, content) VALUES (?, ?)");
                 FileInputStream fis = new FileInputStream(file)) {
                pst.setString(1, file.getName());
                pst.setBinaryStream(2, fis, (int) file.length());
                pst.executeUpdate();
                statusLabel.setText("File uploaded: " + file.getName());
            } catch (SQLException | IOException ex) {
                statusLabel.setText("Error: " + ex.getMessage());
            }
        }
    }

    private void retrieveFile() {
        try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);
             Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery("SELECT * FROM Files ORDER BY id DESC LIMIT 1")) {
            if (rs.next()) {
                byte[] fileContent = rs.getBytes("content");
                String fileName = "retrieved_" + rs.getString("name");
                try (FileOutputStream fos = new FileOutputStream(fileName)) {
                    fos.write(fileContent);
                    statusLabel.setText("File retrieved and saved as: " + fileName);
                }
            } else {
                statusLabel.setText("No file found.");
            }
        } catch (SQLException | IOException ex) {
            statusLabel.setText("Error: " + ex.getMessage());
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(FileStorageApp::new);
    }
}
```



Look In: user

File Name:

Files of Type: All Files