**JAMIA MILLIA ISLAMIA**

JAVA ASSIGNMENT

COURSE CODE : CA18 LAB II (JAVA)

SUBMITTED BY : PIYUSH SOLANKI

MCA , SEMESTER 1

STUDENT ID : 202404636

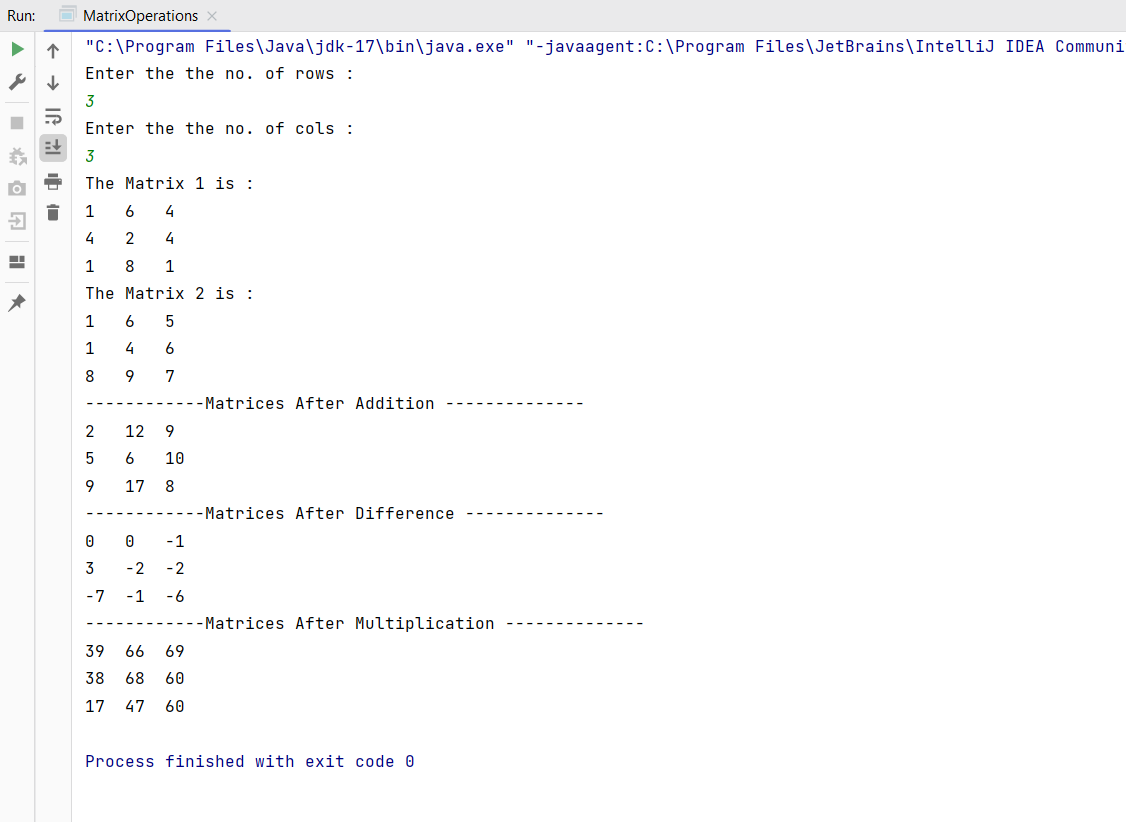
SUBMITTED TO : DR. KHALID RAZA

DEPARTMENT OF COMPUTER SCIENCE

1. Write a Java Program for Matrix Operations . Take two matrices as input of similar dimensions , populate the values randomly and perform sum , difference and multiplication.

import java.util.Random;  
import java.util.Scanner;  
  
public class MatrixOperations {  
 *//Code to create matrix with random values* private static void createMatrix(int m[][], Random random) {  
 for (int i = 0; i < m.length; i++) {  
 for (int j = 0; j < m[i].length; j++) {  
 m[i][j] = random.nextInt(10);  
 }  
 }  
 }  
  
 *//Code to Display the created Matrix* private static void displayMatrix(int m[][]) {  
 for (int i = 0; i < m.length; i++) {  
 for (int j = 0; j < m[i].length; j++) {  
 System.*out*.print(m[i][j] + "\t");  
 }  
 System.*out*.println();  
 }  
 }  
  
 *//Code to perform addition of two matrices* private static void addMatrices(int m1[][], int m2[][], int res[][]) {  
 for (int i = 0; i < m1.length; i++) {  
 for (int j = 0; j < m1[i].length; j++) {  
 res[i][j] = m1[i][j] + m2[i][j];  
 }  
 }  
 }  
  
 *//Code to perform subtraction of two matrices* private static void subMatrices(int m1[][], int m2[][], int res[][]) {  
 for (int i = 0; i < m1.length; i++) {  
 for (int j = 0; j < m1[i].length; j++) {  
 res[i][j] = m1[i][j] - m2[i][j];  
 }  
 }  
 }  
  
 *//Code to perform product of two matrices* private static void multiplyMatrices(int m1[][], int m2[][], int res[][]) {  
 for (int i = 0; i < m1.length; i++) {  
 for (int j = 0; j < m1[i].length; j++) {  
 res[i][j] = 0;  
 for (int k = 0; k < m1[i].length; k++) {  
 res[i][j] += m1[i][k] \* m2[k][j];  
 }  
 }  
 }  
 }  
  
  
 public static void main(String[] args) {  
 int rows, cols;  
 Random random = new Random();  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.println("Enter the the no. of rows :");  
 rows = scan.nextInt();  
 System.*out*.println("Enter the the no. of cols :");  
 cols = scan.nextInt();  
 int m1[][] = new int[rows][cols];  
 int m2[][] = new int[rows][cols];  
 int add[][] = new int[rows][cols];  
 int diff[][] = new int[rows][cols];  
 int prod[][] = new int[rows][cols];  
  
 *//Creating and Populating the matrices  
 createMatrix*(m1, random);  
 System.*out*.println("The Matrix 1 is : ");  
 *displayMatrix*(m1);  
 *createMatrix*(m2, random);  
 System.*out*.println("The Matrix 2 is : ");  
 *displayMatrix*(m2);  
  
 *//Performing Operations  
 //1. Addition  
 addMatrices*(m1, m2, add);  
 System.*out*.println("------------Matrices After Addition --------------");  
 *displayMatrix*(add);  
  
 *//2. Subtraction  
 subMatrices*(m1, m2, diff);  
 System.*out*.println("------------Matrices After Difference --------------");  
 *displayMatrix*(diff);  
  
 *//3. Multiplication  
 multiplyMatrices*(m1, m2, prod);  
 System.*out*.println("------------Matrices After Multiplication --------------");  
 *displayMatrix*(prod);  
  
 scan.close();  
 }  
  
}

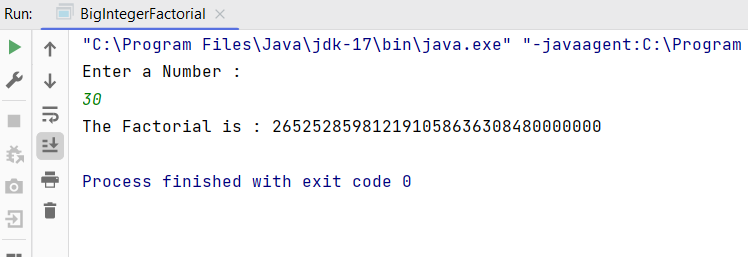
Output :



1. Write a Java Program for calculating the factorial of a number using BigNumbers class in Java.

import java.math.BigInteger;  
import java.util.\*;  
public class BigIntegerFactorial {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Enter a Number : ");  
 int num = scanner.nextInt();  
 BigInteger fact = BigInteger.*ONE*;  
 for(int i=1; i<=num;i++){  
 fact =fact.multiply(BigInteger.*valueOf*(i));  
 }  
   
 System.*out*.println("The Factorial is : "+fact);  
 }  
}

Output :



1. Define a class called ‘Bank Account’ that represents the bank account of a person that comprises of several selected data such as Account Number , First name , Last name , Address , City , Pincode , Pan Number and balance. Introduce necessary constructors for initialisation of data further this class has following methods.

deposit(double amount)

withdraw(double amount)

checkBalance()

transfer(BankAccount ac , double amount)

Use Boolean for deposit and withdraw.

Bank Account Class

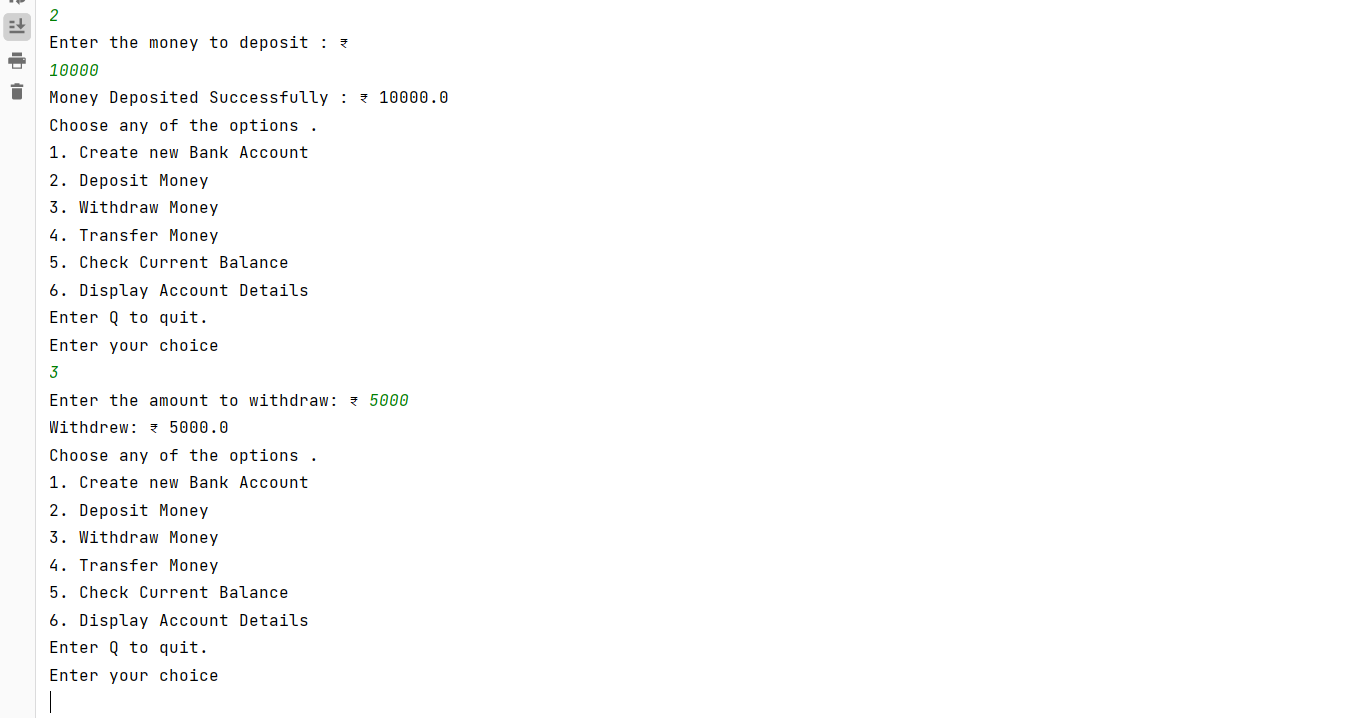
import java.util.Random;  
  
public class BankAccount {  
 private String fname , lname , city , address;  
 private static long *uniqueAccountNo*=1900555500001000L;  
 private long accountNo;  
 private int pinCode;  
 private double balance;  
 private String pan;  
  
*//Constructor to initialize the variables and necessary data .* BankAccount(String fname , String lname , String address , String city , String pan , int pinCode , double balance){  
 this.fname = fname;  
 this.lname = lname;  
 this.address =address;  
 this.city =city;  
 this.pan = pan;  
 this.pinCode = pinCode;  
 this.balance = balance;  
 this.accountNo = setAccountNo();  
 }  
  
 *//Method to generate random and unique account number.* public long setAccountNo(){  
 Random random = new Random();  
 accountNo = *uniqueAccountNo* +(random.nextInt(1000,5000));  
 return accountNo;  
 }  
  
 *//Method to deposit amount .* public boolean deposit(double amount){  
 if(amount >0){  
 this.balance +=amount;  
 return true;  
 }  
 return false;  
 }  
  
 *//Method to withdraw amount .* public boolean withDraw(double amount){  
 if(this.balance>0 && this.balance >=amount){  
 this.balance -= amount;  
 return true;  
 }  
 return false;  
 }  
  
 *// Transfer money to another account* public boolean transfer(BankAccount recipient, double amount) {  
 if (amount > 0 && amount <= balance) {  
 if (this.withDraw(amount)) {  
 recipient.deposit(amount);  
 return true;  
 }  
 }  
 return false;  
 }  
  
 *//Method to check balance* public double checkBalance(){  
 return balance;  
 }  
  
 *//Method to display the bank account details* public void displayAccountDetails(){  
 System.*out*.println("----------------------------------------------------------------------------");  
 System.*out*.println("---------------Account Details---------------");  
 System.*out*.println("Account Number : "+accountNo);  
 System.*out*.println("First Name : "+fname +"\t"+"Last Name : "+lname);  
 System.*out*.println("Address : "+address);  
 System.*out*.println("City : "+city);  
 System.*out*.println("Pincode : "+pinCode);  
 System.*out*.println("Pan Number : "+pan);  
 System.*out*.println("----------------------------------------------------------------------------");  
 }  
  
  
}

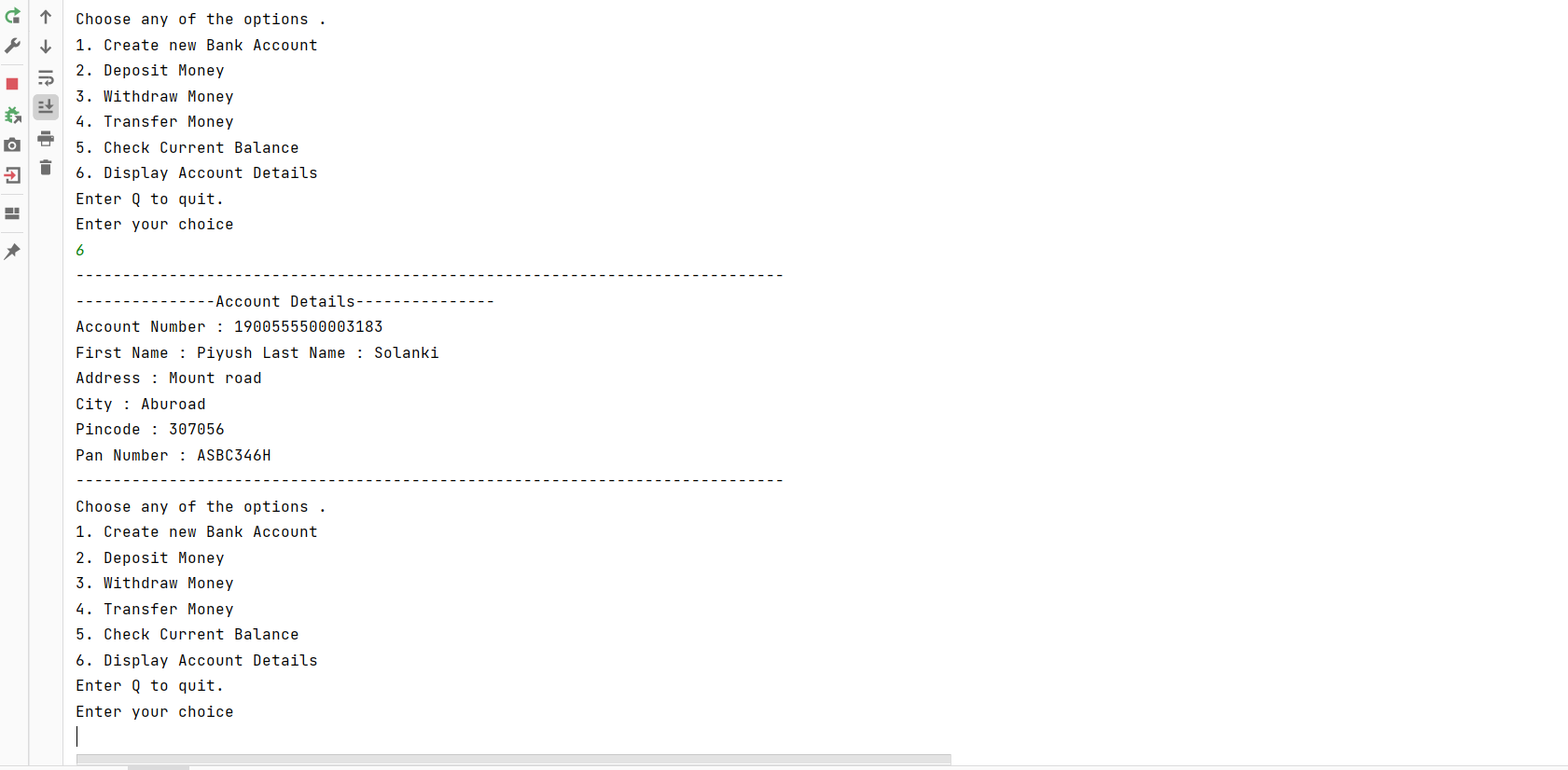
Tester class

import java.util.Scanner;  
  
public class Tester {  
 public static void main(String[] args) {  
 System.*out*.println("-------------------------------------------------------------------------------------");  
 System.*out*.println("Welcome to Bank of India");  
 Scanner scanner = new Scanner(System.*in*);  
 String fname , lname , address , city,pan;  
 int pinCode;  
 double balance,depositAmount,withdrawlAmount;  
 BankAccount B=null , recipientAccount =null;  
 char ch;  
 do{  
 System.*out*.println("Choose any of the options .");  
 System.*out*.println("1. Create new Bank Account\n"+"2. Deposit Money\n"+"3. Withdraw Money\n"+"4. Transfer Money\n"+"5. Check Current Balance\n"+"6. Display Account Details");  
 System.*out*.println("Enter Q to quit.");  
 System.*out*.println("Enter your choice");  
 ch = scanner.next().charAt(0);  
 switch (ch){  
 case '1':  
 scanner.nextLine();  
 System.*out*.println("Enter your First name : ");  
 fname=scanner.nextLine();  
 System.*out*.println("Enter your Last name : ");  
 lname=scanner.nextLine();  
 System.*out*.println("Enter your Address : ");  
 address= scanner.nextLine();  
 System.*out*.println("Enter your City : ");  
 city = scanner.nextLine();  
 System.*out*.println("Enter your Pincode : ");  
 pinCode = scanner.nextInt();  
 System.*out*.println("Enter your Pan Details : ");  
 pan = scanner.next();  
 System.*out*.println("Enter your balance : ₹ ");  
 balance = scanner.nextDouble();  
 B = new BankAccount(fname,lname,address,city,pan,pinCode,balance);  
 System.*out*.println("Congratulations!! Account Created Successfully.");  
 break;  
 case '2':  
 if(B != null){  
 System.*out*.println("Enter the money to deposit : ₹");  
 depositAmount = scanner.nextDouble();  
 if(B.deposit(depositAmount)){  
 System.*out*.println("Money Deposited Successfully : ₹ "+depositAmount);  
 }  
 else {  
 System.*out*.println("Invalid deposit amount.");  
 }  
 }  
 else {  
 System.*out*.println("No account found. Please create an account first.");  
 }  
 break;  
 case '3':  
 if (B != null) {  
 System.*out*.print("Enter the amount to withdraw: ₹ ");  
 withdrawlAmount = scanner.nextDouble();  
 if (B.withDraw(withdrawlAmount)) {  
 System.*out*.println("Withdrew: ₹ " + withdrawlAmount);  
 } else {  
 System.*out*.println("Invalid withdrawal amount or insufficient balance.");  
 }  
 } else {  
 System.*out*.println("No account found. Please create an account first.");  
 }  
 break;  
 case '4':  
 if (B != null) {  
 if (recipientAccount == null) {  
 System.*out*.println("Recipient account not initialized. Creating a default recipient account.");  
 recipientAccount = new BankAccount("Raj", "Kumar", "11 Hudson Lane", "New Delhi", "XYZ123456", 654321, 5000.00);  
 }  
  
 System.*out*.print("Enter the amount to transfer: ₹");  
 double transferAmount = scanner.nextDouble();  
 scanner.nextLine();  
 if (B.transfer(recipientAccount, transferAmount)) {  
 System.*out*.println("Transferred: ₹" + transferAmount + " to recipient account.");  
 } else {  
 System.*out*.println("Insufficient balance or invalid transfer amount.");  
 }  
 } else {  
 System.*out*.println("No account found. Please create an account first.");  
 }  
 break;  
 case '5':  
 if(B!=null){  
 double curr\_Balance = B.checkBalance();  
 System.*out*.println("Current Balance is : ₹ "+curr\_Balance);  
 }else {  
 System.*out*.println("No account found. Please create an account first.");  
 }  
 break;  
 case '6':  
 if (B != null) {  
 B.displayAccountDetails();  
 } else {  
 System.*out*.println("No account found. Please create an account first.");  
 }  
 break;  
  
 case 'Q':  
 case 'q':  
 System.*out*.println("Exiting...");  
 scanner.close();  
 System.*exit*(0);  
 break;  
 default:  
 System.*out*.println("Invalid choice. Please select a valid option.");  
 break;  
 }  
 }while(ch!='Q' && ch!='q');  
  
 }  
}

Output :









1. a) Define a class ‘Student’ that has necessary methods. The class also has a data member called static instance count which stores the total number of instance of student class that has been made. Write a tester class to demonstrate the working of it.

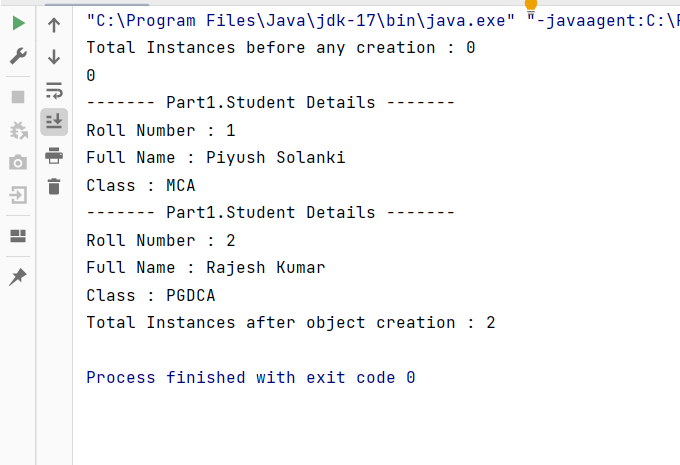
Student class

public class Student {  
  
 private int rollno;  
 private String fullName;  
 private String className;  
 private static int *instanceCount* = 0;  
  
 Student(int rollno, String fullName, String className) {  
 *instanceCount*++;  
 this.rollno = rollno;  
 this.fullName = fullName;  
 this.className = className;  
 }  
  
 public void getDetails() {  
 System.*out*.println("------- Part1.Student Details -------");  
 System.*out*.println("Roll Number : " + rollno);  
 System.*out*.println("Full Name : " + fullName);  
 System.*out*.println("Class : " + className);  
 }  
  
 public static int getInstanceCount() {  
 return *instanceCount*;  
 }  
}

Main class

public class Main{  
 public static void main(String[] args) {  
 System.*out*.println("Total Instances before any creation : "+ Student.*getInstanceCount*());  
 System.*out*.println(Student.*getInstanceCount*());  
 Student s1 = new Student(1, "Piyush Solanki", "MCA");  
*// System.out.println(Part1.Student.getInstanceCount());* s1.getDetails();  
 Student s2 = new Student(2, "Rajesh Kumar", "PGDCA");  
 s2.getDetails();  
*// System.out.println(Part1.Student.getInstanceCount());* System.*out*.println("Total Instances after object creation : "+ Student.*getInstanceCount*());  
 }  
}

Output :



b) Define a Singleton class. The advantage of this class is that it will allow the create of only one object. Write a tester program to demonstrate the working.

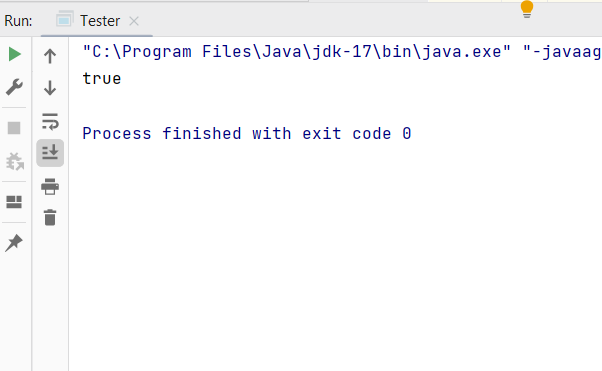
Singleton class

public class Singleton {  
 private static Singleton *instance* = null;  
 private Singleton(){  
 }  
  
 public static Singleton getInstance(){  
 if(*instance* == null){  
 *instance* = new Singleton();  
 }  
 return *instance*;  
 }  
  
}

Tester class

public class Tester {  
 public static void main(String[] args) {  
 Singleton s1 = Singleton.*getInstance*();  
 Singleton s2 = Singleton.*getInstance*();  
 System.*out*.println(s1==s2);  
 }  
}

Output :



1. Define your own package myjava.myMath . The myMath package should contain following classes.

Area

Factorial

Volume

Area class in the package myjava.myMath

package myJava.myMath;  
  
public class Area {  
 *//Function to calculate area of Circle* public static double area(double r){  
 return (Math.*PI* \* r \* r);  
 }  
  
 *//Function to calculate area of rectangle* public static double area(double l , double b){  
 return l \* b;  
 }  
  
 *//Function to calcuate surface area of cylinder* public static double areaCylinder(double r , double h){  
 double result = (2\**area*(r))+(2\*Math.*PI*\*r\*h);  
 return result;  
 }  
  
}

Factorial class in myjava.myMath package

package myJava.myMath;  
import java.math.BigInteger;  
  
public class Factorial {  
 *//Factorial function to calculate for small numbers* public static int fact(int n){  
 int fact =1;  
 for(int i=1;i<=n;i++){  
 fact \*=i;  
 }  
 return fact;  
 }  
  
 *//Factorial function to handle BigNumbers* public static BigInteger factB(int n){  
 BigInteger fact= BigInteger.*ONE*;  
 for(int i=1;i<=n;i++){  
 fact = fact.multiply(BigInteger.*valueOf*(i));  
 }  
 return fact;  
 }  
  
}

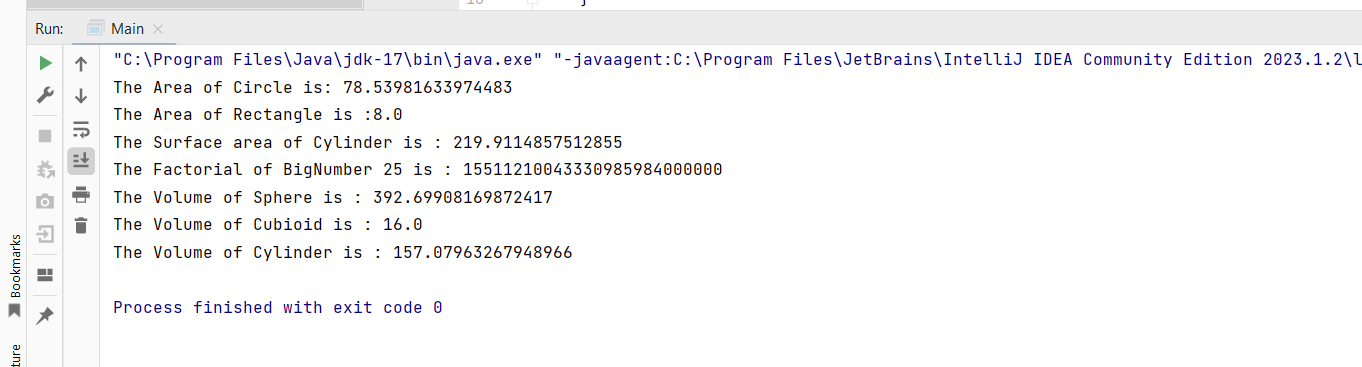
Volume class in myjava.myMath package

package myJava.myMath;  
  
public class Volume {  
 *//Function to calculate volume of sphere* public static double vol(double r){  
 return (4/3\*Math.*PI*\*r\*r\*r);  
 }  
  
 *//Function to calculate volume of cuboid* public static double vol(double a, double b, double c){  
 return a\*b\*c;  
 }  
  
 *//Function to calculate volume of cylinder* public static double vol(double r , double h){  
 return (Math.*PI* \*r\*r\*h);  
 }  
}

Main class to check the program

import myJava.myMath.Area;  
import myJava.myMath.Factorial;  
import myJava.myMath.Volume;  
  
public class Main {  
 public static void main(String[] args) {  
 int r =5 , l=2 , b=4 , h=2;  
 System.*out*.println("The Area of Circle is: "+ Area.*area*(r));  
 System.*out*.println("The Area of Rectangle is :"+Area.*area*(l,b));  
 System.*out*.println("The Surface area of Cylinder is : "+Area.*areaCylinder*(r,h));  
 System.*out*.println("The Factorial of BigNumber 25 is : "+ Factorial.*factB*(25));  
 System.*out*.println("The Volume of Sphere is : "+ Volume.*vol*(r));  
 System.*out*.println("The Volume of Cubioid is : "+Volume.*vol*(l,b,h));  
 System.*out*.println("The Volume of Cylinder is : "+Volume.*vol*(r,h));  
  
 }  
}

Output :



## 6.Write a Java program that takes a paragraph of text as input and does the following

## i) Count the lines , words and characters (ii) Make entire sentence in sentence case (iii) Reverse words (iv) Reverse the entire paragraph.

MyFrame.java

import javax.swing.\*;  
import java.awt.\*;  
  
public class MyFrame extends JFrame {  
 private JButton b1, b2, b3, b4, b5, b6, b7;  
 private JTextArea ta, result;  
  
  
 public MyFrame() {  
 setTitle("String Manipulations");  
 setDefaultCloseOperation(*EXIT\_ON\_CLOSE*);  
 setSize(500, 500);  
 setLocationRelativeTo(null);  
 setResizable(false);  
 setLayout(new BorderLayout(10, 10));  
  
 *// Create the main panel* JPanel mainPanel = new JPanel();  
 mainPanel.setLayout(new GridBagLayout());  
 GridBagConstraints gbc = new GridBagConstraints();  
 gbc.insets = new Insets(5, 5, 5, 5);  
  
 *// Add label* gbc.gridx = 0;  
 gbc.gridy = 0;  
 gbc.gridwidth = 2;  
 mainPanel.add(new JLabel("Enter your text for String Manipulations:"), gbc);  
  
 *// Add text area* ta = new JTextArea(8, 30);  
 ta.setLineWrap(true);  
 JScrollPane scrollPane = new JScrollPane(ta);  
 gbc.gridy = 1;  
 mainPanel.add(scrollPane, gbc);  
  
 *// Create button panel* JPanel buttonPanel = new JPanel();  
 buttonPanel.setLayout(new GridLayout(4, 2, 5, 5));  
 b1 = new JButton("Count Words");  
 b2 = new JButton("Count Characters");  
 b3 = new JButton("Count Lines");  
 b4 = new JButton("Sentence Case");  
 b5 = new JButton("Reverse Words");  
 b6 = new JButton("Reverse Text");  
 b7 = new JButton("Reset Text");  
  
 *// Add buttons to button panel* buttonPanel.add(b1);  
 buttonPanel.add(b2);  
 buttonPanel.add(b3);  
 buttonPanel.add(b4);  
 buttonPanel.add(b5);  
 buttonPanel.add(b6);  
 buttonPanel.add(b7);  
  
 *// Add button panel to main panel* gbc.gridy = 2;  
 gbc.gridwidth = 2;  
 mainPanel.add(buttonPanel, gbc);  
  
 *// Add result label* gbc.gridy = 3;  
 mainPanel.add(new JLabel("Your Result:"), gbc);  
  
 *// Add result text area* result = new JTextArea(5, 30);  
 result.setEnabled(false);  
 JScrollPane resultScrollPane = new JScrollPane(result); *// Add scroll pane* gbc.gridy = 4;  
 mainPanel.add(resultScrollPane, gbc);  
  
 *// Add main panel to the frame* add(mainPanel, BorderLayout.*CENTER*);  
  
 *// Add action listeners for buttons* addActionListeners();  
  
 *// Set frame visibility* setVisible(true);  
 }  
  
 private void addActionListeners() {  
 b1.addActionListener(e -> {  
 int words = new StringFunctions(ta.getText()).getWordsCount();  
 result.setText("Words Count: " + words);  
 });  
  
 b2.addActionListener(e -> {  
 int chars = new StringFunctions(ta.getText()).getCharactersCount();  
 result.setText("Characters Count: " + chars);  
 });  
  
 b3.addActionListener(e -> {  
 int lines = new StringFunctions(ta.getText()).getLines();  
 result.setText("Lines Count: " + lines);  
 });  
  
 b4.addActionListener(e -> {  
 String sen = new StringFunctions(ta.getText()).makeSentenceCase();  
 result.setText(sen);  
 });  
  
 b5.addActionListener(e -> {  
 String newStr = new StringFunctions(ta.getText()).reverseWords();  
 result.setText(newStr);  
 });  
  
 b6.addActionListener(e -> {  
 StringBuilder txt = new StringFunctions(ta.getText()).reverseString();  
 result.setText(String.*valueOf*(txt));  
 });  
  
 b7.addActionListener(e -> {  
 ta.setText("");  
 result.setText("");  
 });  
 }  
}

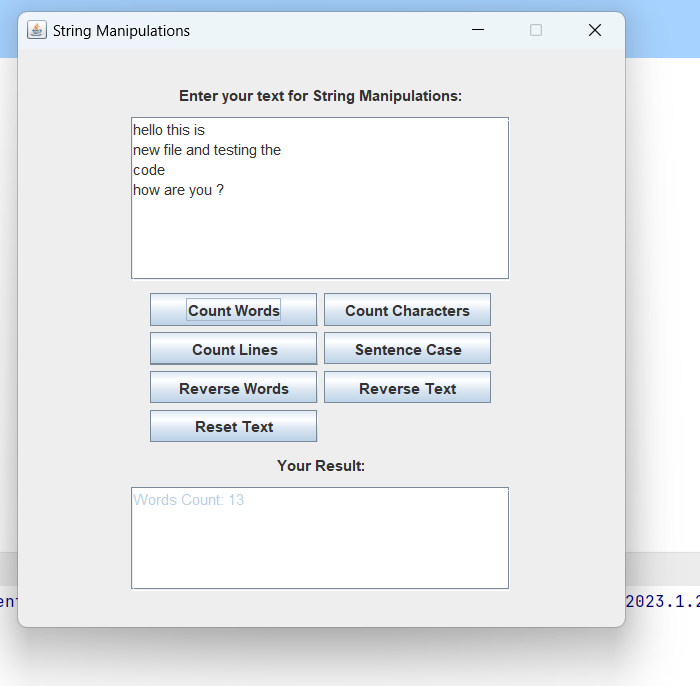
StringFunctions class

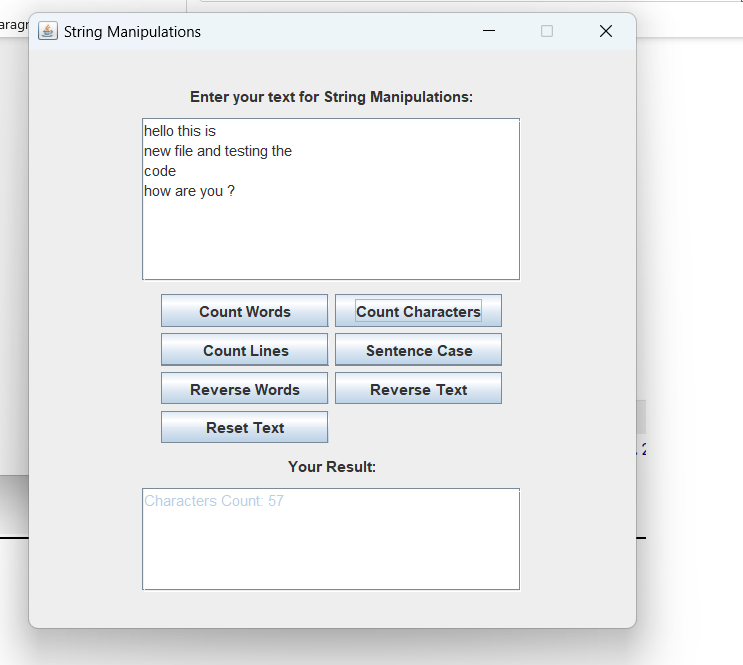
public class StringFunctions {  
 private String text;  
  
 *//Constructor to initialise the String* public StringFunctions(String text){  
 setText(text);  
 }  
  
 *//Method to set the text* public void setText(String text){  
 this.text = text;  
 }  
  
 *//Method to get the count of characters in the String.* public int getCharactersCount(){  
 if(text.isEmpty()|| text==null)  
 return 0;  
 return getText().length();  
 }  
  
 *//Method to get the words count in the String.* public int getWordsCount(){  
 if(text== null || text.isEmpty()){  
 return 0;  
 }  
 String words[] = text.trim().split("\\s+");  
 return words.length;  
 }  
  
 *//Method to get the count of number of lines in the String.* public int getLines(){  
 if(text== null || text.isEmpty()){  
 return 0;  
 }  
 int no\_of\_lines=0;  
 String lines[] = getText().split("\\r?\\n");  
 no\_of\_lines = lines.length;  
 return no\_of\_lines;  
 }  
  
 *//Method to reverse each word in the given String.* public String reverseWords(){  
 String [] words= text.split(" ");  
 String [] newWords = new String[words.length];  
 *//Intialising all the words to empty String* for(int i=0;i< newWords.length;i++){  
 newWords[i]="";  
 }  
 *//Outer Loop to get and traverse each word in the string.* for(int i=0;i<words.length;i++){  
 *//Character Array to store characters of each word in the given string* char []word = words[i].toCharArray();  
 *//Loop to add the reverse word in the newWords array.* for(int j=word.length-1;j>=0;j--){  
 newWords[i]+=(word[j]);  
 }  
 }  
 *//Finally Joining the array back to produce the original string.* String newText = String.*join*(" ",newWords);  
 return newText;  
 }  
 public String getText(){  
 return this.text;  
 }  
  
 *//Method to make the String in the sentence case.* public String makeSentenceCase(){  
 String []lines = text.split("\\r?\\n");  
 StringBuilder newLines = new StringBuilder();  
 for (String line : lines) {  
 if (!line.trim().isEmpty()) {  
 newLines.append(Character.*toUpperCase*(line.charAt(0)))  
 .append(line.substring(1).toLowerCase())  
 .append("\n");  
 }  
 }  
 return newLines.toString().trim();  
 }  
  
 *//Method to print the entire String in the reverse order* public StringBuilder reverseString(){  
 StringBuilder newStr = new StringBuilder();  
 if(text.isEmpty() || text == null){  
 return new StringBuilder(" ");  
 }  
 String []words = getText().split(" ");  
 for(int i=words.length-1;i>=0;i--){  
 newStr.append(words[i]);  
 if (i > 0) {  
 newStr.append(" ");  
 }  
 }  
 return newStr;  
 }  
}

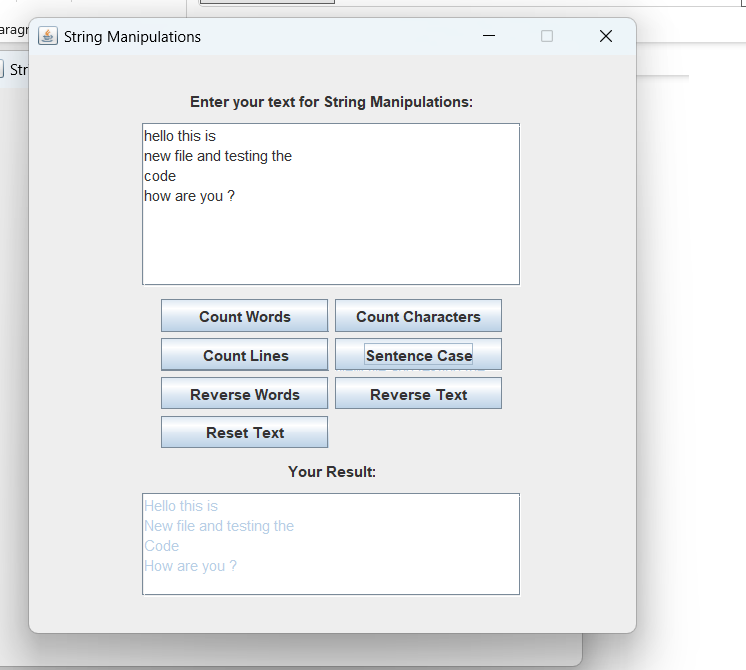
Test Class

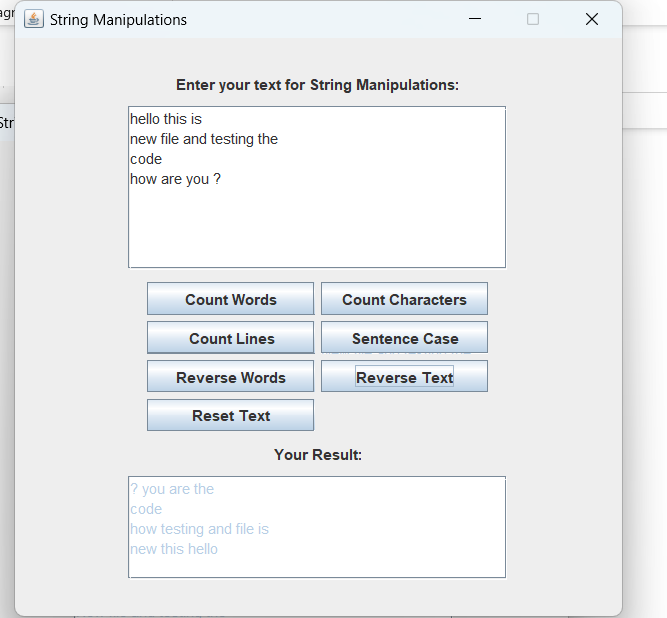
public class Test {  
 public static void main(String[] args) {  
 MyFrame f1 = new MyFrame();  
 }  
}

Output









## 7.Write a java program to generate random password of length 8-15 characters with following rule (i)Must have letters , special characters and numbers (ii)At least one character in uppercase

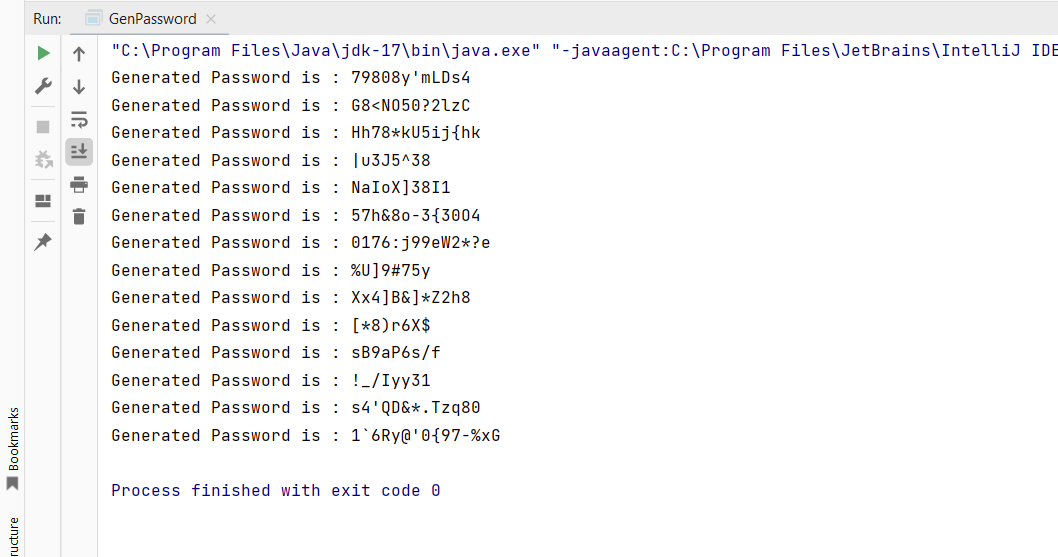
Password class

import java.util.Random;  
  
public class Password {  
 *//Declarations of all the sets needed for password.* private static final char[] *specialChars* = {'!','\"','#','$','%','&','\'','(',')','\*','+','-','.','/',':',';','<','=','>','?','@','[',']','^','\_',  
 '`','{','|','}','~'};  
 private static final char[] *lowerLetters* ={'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};  
 private static final char[] *upperLetters* ={'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};  
 private static final char[] *digits*={'0','1','2','3','4','5','6','7','8','9'};  
  
 private Random random = new Random();  
  
 public String generatePassword(){  
 StringBuilder password = new StringBuilder();  
 password.append(*specialChars*[random.nextInt(*specialChars*.length)]);  
 password.append(*lowerLetters*[random.nextInt(*lowerLetters*.length)]);  
 password.append(*upperLetters*[random.nextInt(*upperLetters*.length)]);  
 password.append(*digits*[random.nextInt(*digits*.length)]);  
  
 int length = random.nextInt(8,16);  
 char[][] chars = new char[][]{*specialChars*,*lowerLetters*,*upperLetters*,*digits*};  
  
 for(int i=4;i<length;i++){  
 int index = random.nextInt(0,4);  
 int el;  
 if(index == 0){  
 el= random.nextInt(*specialChars*.length);  
 password.append(chars[index][el]);  
 }  
 if(index ==1){  
 el= random.nextInt(*lowerLetters*.length);  
 password.append(chars[index][el]);  
 }  
 if(index ==2){  
 el= random.nextInt(*upperLetters*.length);  
 password.append(chars[index][el]);  
 }  
 if(index == 3){  
 el= random.nextInt(*digits*.length);  
 password.append(chars[index][el]);  
 }  
  
 }  
  
 return "Generated Password is : "+shuffle(password.toString());  
 }  
  
 *//Shuffling the password to ensure the random occuring of characters* private String shuffle(String pass){  
 char[] chars = pass.toCharArray();  
 for(int i=chars.length-1;i>0;i--){  
 int j = random.nextInt(i + 1);  
 char temp = chars[i];  
 chars[i] = chars[j];  
 chars[j] = temp;  
 }  
 return new String(chars);  
 }  
}

Test Class

public class GenPassword {  
 public static void main(String[] args) {  
 Password p = new Password();  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
 System.*out*.println(p.generatePassword());  
*// PasswordFrame p = new PasswordFrame();* }  
}

Output



## 8. Define an exception class to validate month and day if the input month is beyond the range of 1-12, an invalid month exception should be generated. Similarly, if the input day is out of 1-31, an invalidDayException should be generated. Test these exceptions in the main class.

InvalidDayException Class

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

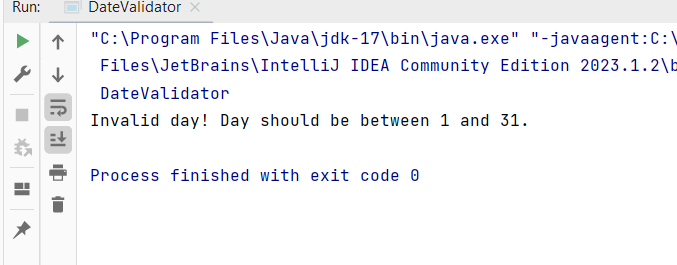
InvalidMonthException Class

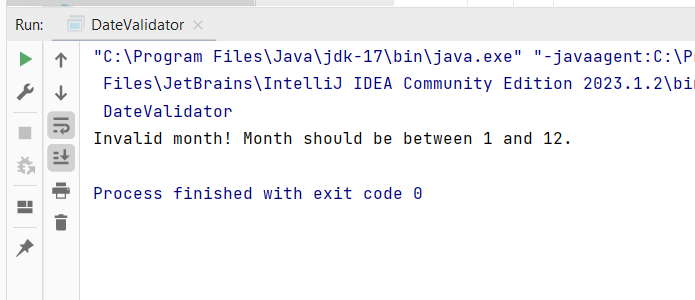
class InvalidMonthException extends Exception {  
 public InvalidMonthException(String message) {  
 super(message);  
 }  
}

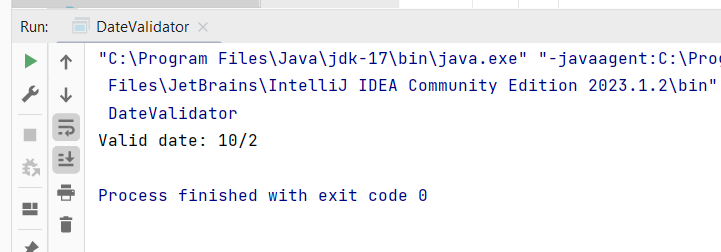
DataValidator Class

public class DateValidator {  
  
 *// Method to validate month* public static void validateMonth(int month) throws InvalidMonthException {  
 if (month < 1 || month > 12) {  
 throw new InvalidMonthException("Invalid month! Month should be between 1 and 12.");  
 }  
 }  
  
 *// Method to validate day* public static void validateDay(int day) throws InvalidDayException {  
 if (day < 1 || day > 31) {  
 throw new InvalidDayException("Invalid day! Day should be between 1 and 31.");  
 }  
 }  
  
 public static void main(String[] args) {  
 int month = 10;  
 int day = 32;  
  
 try {  
  
 *validateMonth*(month);  
 *validateDay*(day);  
  
 System.*out*.println("Valid date: " + month + "/" + day);  
  
 } catch (InvalidMonthException e) {  
 System.*out*.println(e.getMessage());  
 } catch (InvalidDayException e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}

Output





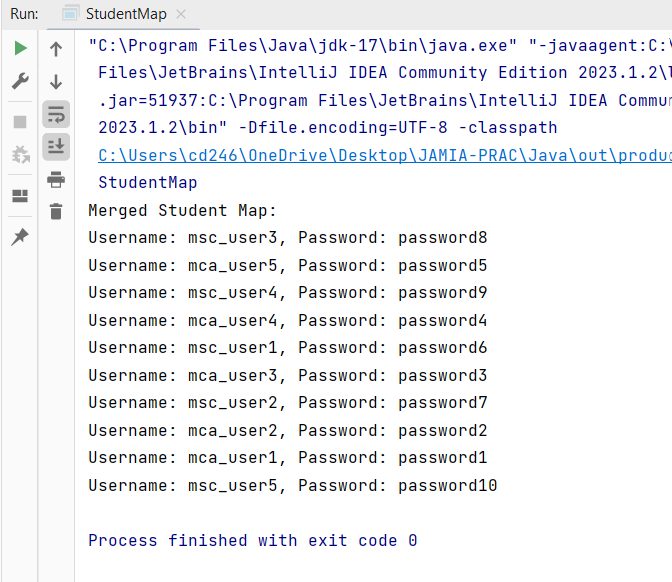


## 9. Write a java program using map suitable to store username and password for the students of MCA and MSC. Add one map to another. Display the content of merged map.

StudentMap Class

import java.util.\*;  
  
public class StudentMap {  
 public static void main(String[] args) {  
 *// Create a Map for MCA students* Map<String, String> mcaStudents = new HashMap<>();  
 mcaStudents.put("mca\_user1", "password1");  
 mcaStudents.put("mca\_user2", "password2");  
 mcaStudents.put("mca\_user3", "password3");  
 mcaStudents.put("mca\_user4", "password4");  
 mcaStudents.put("mca\_user5", "password5");  
  
 *// Create a Map for MSC students* Map<String, String> mscStudents = new HashMap<>();  
 mscStudents.put("msc\_user1", "password6");  
 mscStudents.put("msc\_user2", "password7");  
 mscStudents.put("msc\_user3", "password8");  
 mscStudents.put("msc\_user4", "password9");  
 mscStudents.put("msc\_user5", "password10");  
  
 *// Merge both maps into a single map* Map<String, String> allStudents = new HashMap<>(mcaStudents);  
 allStudents.putAll(mscStudents);  
  
 *// Display the merged map* System.*out*.println("Merged Student Map:");  
 for (Map.Entry<String, String> entry : allStudents.entrySet()) {  
 System.*out*.println("Username: " + entry.getKey() + ", Password: " + entry.getValue());  
 }  
 }  
}

Output



## 10. Write a java program to implement a Generic Array which can store elements of various types . Create a menu driven program to add various functionalities.

Generic Array class

import java.util.Arrays;  
@SuppressWarnings("unchecked")  
  
public class GenericArray<T> {  
 private final int INITIAL\_CAPACITY = 6;  
 private T[] genericArray;  
 private int index;  
  
 public GenericArray() {  
 genericArray = (T[]) new Object[INITIAL\_CAPACITY];  
 index = 0;  
 }  
  
 public void resizeMax(){  
 T[] tempArr = (T[]) new Object[genericArray.length\*2];  
  
 for(int i = 0;i < index;i++){  
 tempArr[i] = genericArray[i];  
 }  
 genericArray = tempArr;  
 }  
  
 private void resizeMin() {  
 T[] tempArr = (T[]) new Object[genericArray.length/2];  
  
 for(int i = 0;i < index;i++){  
 tempArr[i] = genericArray[i];  
 }  
 genericArray = tempArr;  
 }  
  
 public void add(T element){  
 if(index == genericArray.length){  
 resizeMax();  
 }  
 genericArray[index++] = element;  
 }  
  
 public void remove(int position){  
 T[] tempArr = (T[]) new Object[genericArray.length];  
  
 try{  
 if(genericArray[position] == null) {  
 System.*out*.println("No element present at " + position + " index.");  
 return;  
 }  
  
 int k = 0;  
 for(int i = 0;i < genericArray.length;i++){  
 if(i == position){  
 continue;  
 }  
 tempArr[k++] = genericArray[i];  
 }  
 genericArray = tempArr;  
  
 --index;  
 if(index < (genericArray.length/2)){  
 resizeMin();  
 }  
 }  
 catch (ArrayIndexOutOfBoundsException e){  
 System.*out*.println(e.getMessage());  
 }  
 }  
  
 public void search(T element){  
 boolean flag = false;  
 for(int i = 0;i < genericArray.length;i++){  
 if(genericArray[i].equals(element)){  
 System.*out*.println("Element found at index "+i+".");  
 flag = true;  
 break;  
 }  
 }  
 if(!flag){  
 System.*out*.println(element+" not found in GenericArray.");  
 }  
 }  
  
 public int size(){  
 return index;  
 }  
  
 @Override  
 public String toString(){  
 return Arrays.*toString*(Arrays.*copyOf*(genericArray,index));  
 }  
}

Main class

import java.util.Scanner;  
  
public class GMain {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 GenericArray<Object> ga = new GenericArray<>();  
 String ans = "y";  
  
 while(ans.equals("y") || ans.equals("Y")){  
 System.*out*.println("""  
 \n  
 Enter your Choice:  
 1. Add element  
 2. Remove element  
 3. Search element  
 4. Size of array  
 5. Display array  
 \n""");  
  
 int choice;  
 System.*out*.print("Enter your choice: ");  
 choice = sc.nextInt();  
 sc.nextLine();  
  
 switch (choice){  
 case 1:  
 System.*out*.print("Enter an element: ");  
 Object element = sc.next();  
 sc.nextLine();  
 ga.add(element);  
 System.*out*.println("Element Added !");  
 break;  
 case 2:  
 System.*out*.print("Enter the index of element: ");  
 int index = sc.nextInt();  
 sc.nextLine();  
 ga.remove(index);  
 System.*out*.println("Element Removed !");  
 break;  
 case 3:  
 System.*out*.print("Enter the element you want to search ? ");  
 Object ele = sc.next();  
 sc.nextLine();  
 ga.search(ele);  
 System.*out*.println("Searching is complete.");  
 break;  
 case 4:  
 System.*out*.print("Size of array is: "+ga.size()+"\n");  
 break;  
 case 5:  
 System.*out*.println("Generic Array: ");  
 System.*out*.println(ga);  
 break;  
 default:  
 System.*out*.println("You entered wrong choice.");  
 }  
  
 System.*out*.print("Do you want to continue ? ");  
 ans = sc.nextLine();  
 if(ans.equals("n") || ans.equals("N")){  
 System.*out*.println("Exited !");  
 break;  
 }  
 }  
 }  
}

Output

