BHARATI VIDYAPEETH'S

INSTITUTE OF COMPUTER APPLICATIONS & MANAGEMENT

(Affiliated to Guru Gobind Singh Indraprastha University,

Approved by AICTE, New Delhi)

Object-Oriented Programming and Java (MCA-167) Practical File

Submitted To:

Dr. Ritika Wason

(Associate Professor)

Submitted By:

Abhishek Sharma (03911604421)

MCA 1st Sem, Sec 1

INDEX

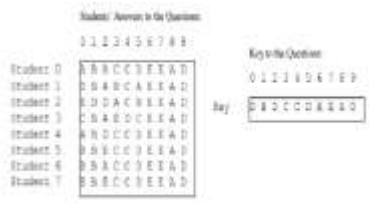
AP1	Explore the basic java program development scenario in Notepad++ and cmd by creating an Integer Adder. The adder prints sum of 5 integer numbers without using single variable where input will be taken through command line arguments. a) Perform the above code using a function and call it in main(). b) Make another class and a function in it to perform the above task.
AP2	Develop a Number Reciprocator java application to computes the sum of the reciprocals with using a single variable in the format: 1/1 + 1/2 + 1/3 + ... + 1/10
AP3	Demonstrate type conversion in a simple java program by casting and checking output in the following cases:- a) Conversion of int to byte b) Conversion of double to int c) Conversion of double to byte d) Conversion of int to char e) Conversion of float to short

AP4	Construct a character counter that inputs a piece of text that is analyzed character by character to determine the vowels, spaces and letters used. Fill in the code that computes the number of spaces, vowels, and consonants. public class StringCharacters { public static void main(String[] args) { String text = "To be or not to be, that is the question;" +"Whether this nobler in the mind to suffer" +" the slings and arrows of outrageous fortune," +" or to take arms against a sea of troubles," +" and by opposing end them?"; int spaces = 0, vowels = 0, letters = 0; **//YOUR CODE HERE** System.out.println("The text contained vowels: " + vowels + "\n" + consonants " + (letters - vowels) + "\n"+ spaces: " + spaces); } }
AP5	Construct a number generator to accept three digits (i.e. 0 - 9) and print all its possible combinations. (For example if the three digits are 1, 2, 3 than all possible combinations are: 123, 132,213, 231, 312, 321.)
AP6	A java standalone application makes use of a parameterized method inside a class. Take the following case: Create a class Box and define a method in this class which will return the volume of the box. Initialize two objects for your class and print out the volumes respectively.
AP7	A java standalone application reads in a sentence from the user and prints it out with each word reversed, but with the words and punctuation in the

	original order.
AP 8	Develop an employee pay generator that works on the following rules- 1. An employee gets paid (hours worked) × (base pay), for each hour up to 40 hours. 2. For every hour over 40, they get overtime = (base pay) × 1.5. 3. The base pay must not be less than the minimum wage ($8.00 an hour). 4. If it is, print an error. If the number of hours is greater than 60, print an error message.//System.err.println();
AP 9	A Financial Calculator to calculate the SimpleInterest and CompoundInterest by taking command line values for principal, rate and time. 1. Extend the code to calculate 'Final Value' of investment (V) of an investment (principal P) compounded yearly for T years at interest rate R is given by the formula: $V = P\ (1 + R)^T$ 2. Perform the above code using a function and call it in main(). Make another class and a function in it to perform the above task.
Advanced Problems:	
A A1	A Gravity Calculator in java to compute the position of an object after falling for 10 seconds, outputting the position in meters. The formula in Math notation is: $x(t) = 0.5 \times a t^2 + v_i t + x_i$ Variable Meaning Value a Acceleration (m/s 2) -9.81 t Time (s) 10 v_i Initial velocity (m/s) 0

	xi Initial position 0 Implement the same using:- 1. Using if-else ladder 2. Using Switch case NOTE- use minimum variables in code and the program should be menu driven that exits only when the user opts to.
AA2	A java application 'MyNumber.java" that performs following operations on a variable 'num' of type double. Code to include the following functionality: 1. Finds the round value of 'num' and stores the result in a variable numRound of type double. 2. Finds the ceil value of 'num' and stores the result in a variable numCeil of type double. 3. Finds the floor value of 'num' and stores the result in a variable numFloor of type double Cast 'num' to type int and stores the result in a variable numInteger of type int.
PQ5	Create an array to hold certain integer elements entered by the user. Search for a given element desired by the user in this array using Linear Search?

BP1	A group of BVICAM friends decide to run the Airtel Delhi Half Marathon. Their names and times (in minutes) are below: Name Time (minutes) Elena 341 Thomas 273 Hamilton 278 Suzie 329 Phil 445 Matt 402 Alex 388 Emma 275 John 243 James 334 Jane 412 Find the fastest runner. Print the name and his/her time (in minutes). Optional: Find the second fastest runner. Print the name and his/her time (in minutes).
BP2	Create a class named DuplicateFinder which initializes an array of at least 15 elements. Define appropriate methods to print its elements and calculate duplicate elements if any. It should detail the number of duplicates along with their frequency of occurrence.

BP3	Define a class named VowelFilter which contains a static method named filterVowel(). This method receives a character array as argument and returns another array which contains only the non-vowel characters of the argument array.
BP4	Create a class named GradeExam to grade a multiple choice test. There are 20 students and 10 questions in the test. Each row records a student's answers to the questions, as shown in the following array: Grade the students according to their score in the test? 
BP5	In a GPS navigation system, Given a set of points, the closest-pair problem is to find the two points that are nearest to each other.
BP6	Develop a command line driven code to accept the following city name as argument in the command line and sort them in alphabetic order – City Name = Kolkata, Chennai, Mumbai, Delhi, Bangalore, Ahmedabad.
BP7	Create a Waffle Class having two data members flavor and price, being populated by a parameterized constructor. Create another class WaffleMain to compare the values and object context of Waffle class data members by equals() and ==. Extend the implementation to get desired results by overriding a) equals()

	b) toString()
BP8	Create a POC(Proof of Concept) to demonstrate usage of various functions of String Class like:- a) charAt() b) length() c) contains() d) equals and == e) indexOf() f) split() g) toUpperCase()
BP9	Use ragged array to provide the output given below (Take row count from user). 1 123 1234 12345 123456 1234567 1 12 123 1234 12345 123456 1234567
BA1	Create a stack to maintain plates in a birthday party. Ask the organizer how many plates do they want initially. Maintain track of each plate used and number of plates added. Finally print the number of total number of plates in the stack, plates used and number of plates left in the stack

BA2	Sudoku is a 9 X 9 grid divided into smaller 3 X 3 boxes (also called regions or blocks) . Fill the empty cells, also called free cells, with numbers 1 to 9 so that every row, every column, and every 3 X 3 box contains the numbers 1 to 9
BA3	An application maintains information regarding teams and players of a particular sport. Identify and implement the relation between team and player. Create a program that provides the following options: a) Enter TeamID b) Enter PlayerID When the user enters a valid TeamID the details of the players belonging to that team should be displayed PlayerID the details of the team to which the player belongs should be displayed. There can be multiple players associated with a country.
	Assignment C
PQ9	Create a class ToStringDemo that uses the toString method to convert a double number entered by the user to a String. Use appropriate methods to count the number of digits before and after the decimal point in the input number?
PQ10	Create a class called FileName to isolate the different parts of a FileName taken as an input from the user
PQ11	Create appropriate classes to reverse a palindrome using String class and StringBuilder. Try to compute which solution computes faster?
	PQ12-14
CP1	Design a class called **DecipherCaesarCode** to decipher the Caesar's code. The program shall prompts user for a ciphertext string consisting of mix-case letters only; compute the plaintext;

	and print the plaintext in uppercase. Design the solution with appropriate methods?
CP2	A word that reads the same backward as forward is called a *palindrome*, e.g., "mom", "dad", "racecar", "madam", and "Radar" (case-insensitive). Create a class called **TestPalindromicWord**, that prompts user for a word and prints ""xxx" is|is not a palindrome".
CP3	A java based program which prompts user for the number of students in a class (a non-negative integer), and saves it in an int variable called numStudents. It then prompts user for the grade of each of the students (integer between 0 to 100) and saves them in an int array called grades. The program shall then compute and print the average (in double rounded to 2 decimal places) and minimum/maximum (in int).
CP4	Demonstrate the working of a Static Inner Class through a class Electronics and within it create Static Inner Class Television that has a method cost() which displays cost of television object passed in constructor of Television class. Demonstrate invoking inner class method with outer object when the method cost() is once a :- a) Instance(Non static) method b) Static method
CP5	Simulate a simple banking application. Provide for classes BankAccount. Account will be of two type- Savings and Current. There should be methods to open an account, close an account and perform withdraw, deposit and transfer operations on an account as abstract methods in Account and properly overridden with definition in Account Types. Test classes should instantiate Account Type Classes and provide a menu driven option for operations.
CP6	Create a package called animals,let this package contain an interface called Animal that generalizes the eat and travel task of any animal. Implement the given interface in the same package animals and override the methods appropriately?

CP7	Create a package personpackage. This package contains a class Person with data members to represent firstName and lastName of a person and appropriate methods to read and display the same. Define appropriate class to test the above class outside the above package?
CP8	Create an Array of Student class objects. a) Sort them by roll number using Comparable Interface. b) Perform sort based on alphabetic first name numbering using Comparator
CP9	Create an interface Relatable that compares the size of objects.Any class can implement Relatable if there is some way to compare the relative "size" of objects instantiated from the class. For strings, it could be number of characters; for books, it could be number of pages; for students, it could be weight; and so forth. For planar geometric objects, area would be a good choice while volume would work for three-dimensional geometric objects. All such classes can implement the isLargerThan() method. Create appropriate implementations.
CP10	Create an interface GeoAnalyzer with a constant PI and methods area() and perimeter(). Let Circle, Ellipse and rectangle implement this class. Define a class Geometry that assigns appropriate objects to GeoAnalyzer reference variable.
	Advanced Questions
CA1	Extend the following methods in a class called DateUtil: a) boolean isLeapYear(int year): returns true if the given year is a leap year. A year is a leap year if it is divisible by 4 but not by 100, or it is divisible by 400.

	b) boolean isValidDate(int year, int month, int day): returns true if the given year, month and day constitute a given date. Assume that year is between 1 and 9999, month is between 1 (Jan) to 12 (Dec) and day shall be between 1 and 28|29|30|31 depending on the month and whether it is a leap year. c) int getDayOfWeek(int year, int month, int day): returns the day of the week, where 0 for SUN, 1 for MON, ..., 6 for SAT, for the given date. Assume that the date is valid. d) String toString(int year, int month, int day): prints the given date in the format "xxxday d mmm yyyy", e.g., "Tuesday 14 Feb 2012". Assume that the given date is valid
	Assignment D
DP1	Extend your Gravity Calculator code (Assignment AA1) to handle exceptions through a try-catch finally block. Handle provision for a divide by zero scenarios caught by NumberFormatException. Explicitly invoke this exception in execution and observe the response.
DP2	Further extend your menu-driven Simple Calculator code (Assignment 2) by adding support a Custom Exception code that raises an 'Invalid Numeral' exception each time the user tries to enter any character except a number for calculation. Explicitly invoke this exception in execution and observe the response.
DP3	Create a class StudentRegistrationCheck that checks eligibility of a student for registration. If the student age<12 and marks<200 then the student is not eligible for registration. `Design appropriate solution using exception handling?`

DP4	An ExceptionPOC class is requesting a number between 1 and 10. Run the program again and enter 5.5. Although this number is between 1 and 10, the program will abort. Examine the error message. You should see the word Exception, the method where the exception occurred (main), the class name of the exception (InputMismatchException), as well as the call stack listing the method calls. Add a try/catch block to catch and handle the InputMismatchException exception. Identify the statements that cause the error as well as the portions of the program that depend upon these statements. Enclose these statements within the try block. Follow the try block with the catch block given below. Note, the InputMismatchException class is defined in java.util and must be imported. Also, when the Scanner throws an InputMismatchException, the input token will remain in the buffer so that it can be examined by the program. Complete code by implementing the same using:- a) Throws method declaration b) Throw keyword
DP5	Extend CP5 (Bank application)with appropriate exception handling?
DP6	Create an Exception class InvalidProductException that can be thrown if a user adds an invalid product.
DP7	Compile and run BadThreads.java: `public class BadThreads {` `  static String message;` `private static class CorrectorThread extends Thread {` `public void run() { try {` `sleep(1000);` `} catch (InterruptedException e) {}` `// Key statement 1:`

```
                message = "Mares do eat oats.";
            }}
public static void main(String args[]) throws InterruptedException
{(new CorrectorThread()).start();
message = "Mares do not eat oats.";
 Thread.sleep(2000);
//        Key        statement        2:
System.out.println(message);
       }}
```

The application should print out "Mares do eat oats." Is it guaranteed to always do this? If not, why not? Would it help to change the parameters of the two invocations of Sleep? How would you guarantee that all changes to the message will be visible in the main thread?

DP8

Implement the producer consumer problem using multithreading in java. In computing, the producer–consumer problem (also known as the bounded- buffer problem) is a classic example of a multi-process synchronization problem. The problem describes two processes, the producer and the consumer, which share a common, fixed-size buffer used as a queue.

a) The producer's job is to generate data, put it into the buffer, and start again.

b) At the same time, the consumer is consuming the data (i.e. removing it from the buffer), one piece at a time.

To make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer.

DP9

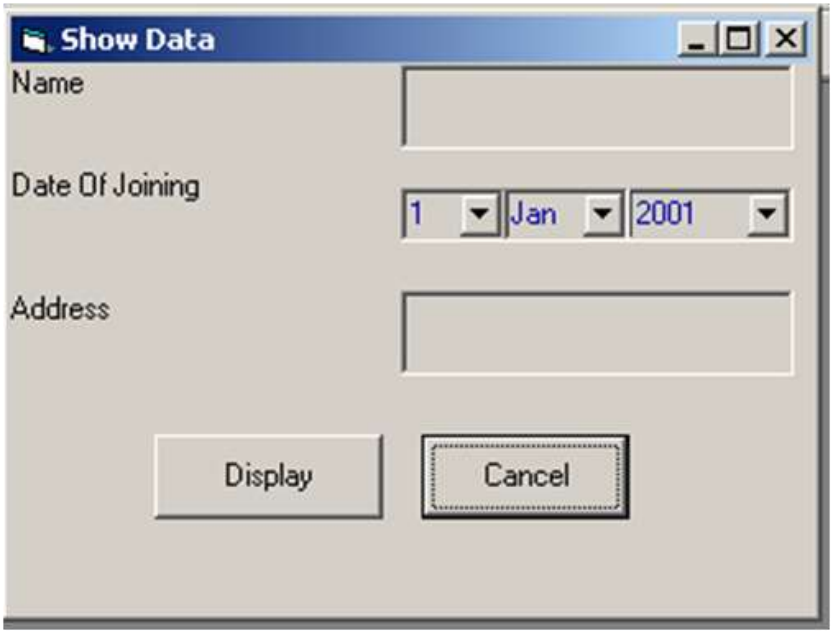
Create a Java application that executes concurrent transactions in a bank?

DP10	Create a list of numbers and then sort in ascending order as well as in descending order simultaneously.
	Assignment E- Collections Framework
EP1	Model Person with name and age.Manage instances of Person by ensuring that no two instances are duplicated?
EP2	Create a subclass of Person (in EP1 above), called ComparablePerson which implements Comparable<Person> interface, and try out the Collections.sort() and Collections.binarySearch() methods on the same.
EP3	Model AddressBookEntry that prints name,address and phone of a person. Allow comparison of AddressBookEntry to compare name in a case sensitive manner?
EP 4	Counts the frequency of each of the words in a file given in the command-line,and save in a map of {word, freq}.
EA1	A java application maintains 5 objects of class Employee in a HashSet. The class has following members: (a) Empno: String (b) Department: String (c) Basic: int (d) HRA_Percent: int

	(e) DA_Percent: int (f) PF_Percent: int Sort and segregate the objects on the basis of Department field. Print the sorted result department wise in the following format: Department: ------ EMPNO BASIC HRA DA PF NET_SALARY ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- Department Average: -----
EA2	A class Student having data members stu_id, stu_name, stu_course. Create 10 objects of this class as desired by the user. Store each object in a sorted order in a hashmap with key as stu_id and value as student object. Create a menu driven code, to provide user options to: a) Add student object in hashmap b) Retrieve any desired object based on roll number entered c) Update details of student object in the map d) Remove student object from map Sort and copy objects in another hashmap

	Assignment F: Networking
FP1	Create a solution to know the address and name of your local machine?
FP2	Create a solution to understand the different components of a URL?
FP3	Create a connection-less client/server application using UDP protocol that sends system date and time in the format requested by the client. a) Client: Reads a string representing the required format from the end-user b) Server:returns the system date and time in the requested format or a default format if received format is not understandable
FP4	Client: Display the required contents. A connection-oriented client/server application using TCP/IP protocol where the client-server has the following responsibilities: a) Server: Creates an Employee class having fields like employeeName, employeeID, and department. Server holds an array of employee objects. b) Client : accepts the employeeID of an employee as an integer from the user. c) Server:Searches for the corresponding employee object, in the array and writes its details to the client stream. d) Client: displays the received object's information

FP5	Create a calculator based client/server application where the client sends request to the server in the form of an arithmetic equation of the form "operand1 operator operator2". The server should respond back to answer the equation?
	Advanced Problems
FA1	Create a real time chatbot application using two protocols. a) Connection-oriented TCP/IP b) Connection-less UDP/IP
	Assignment G: Anonymous inner classes, Event Handling, Swings
GP1	create a class called fruits that has a method mango() that tells if the mango is sweet or sour. Suppose we need a sour mango in taste for only 1 time. Realize this temporary requirement through an anonymous inner class?
GP2	Create a class AWTCounter that starts a counter from 0 and increments its value on every button click?
GP3	Create the following layout using awt/swing



When user clicks the "Display" button the data entered by the user should be displayed in another frame window.

When user clicks the "Cancel" button the data fields should get cleared.

GP4

Create a class MessageBox that extends Frame. The class should have a constructor that takes a String as a parameter to construct a dialog box that displays the message and OK & CANCEL buttons. The dialog box should get closed when the cancel buttons is clicked. Provide some mechanism in the MessageBox class that can be used by the calling program to check which button was pressed by the user. The class should have functions to:

a) Check which button was pressed by the user.

b) Retrieve the string entered by the user, if user pressed OK, null if user pressed CANCEL.

Test this class to get a message from a user and display it on a Frame window.

GP5	Use AWT /Swings to develop a Contact Manager. The GUI application maintains a simple list of contacts storing contact name and number. It provides basic features of CRUD (Create, Update, Retrieve and Delete) as per user choice. The application also supports search feature.
GP6	Create a class MyTextEditor to simulate a notepad using Swings?
	Advanced Problems
GA1	Create a frame that displays a text area in center and textfield in the south region of the frame. The text area is supposed to hold the contents of a text file. The text field is supposed to hold a file name. The frame should have a menu with the following options: File New Open Save ------ Exit When the user clicks New both the input fields should be cleared. When the user clicks Open the contents of the text file whose name is in the text field should be displayed in the text area. When the user clicks Save the contents of the text area should get saved in the file whose name is given in the text field. Clicking Exit should terminate the application.

GA2	frame should have following menu options: Shape Style Color Line Fill Red Rectangle Draw Blue Oval Green The point where the user clicks the left mouse button should be Create a frame that can be used to perform simple drawing operations. The treated as the top-left corner of the shape and the point where the user releases the left mouse button should be treated as the bottom-right corner of the shape.
GA3	Create a frame that displays a pie chart based upon a set of numerical-values entered by the user as command line argument.
GA4	Create a frame that displays a bar chart based upon a set of numerical values entered by the user as command line argument.
	Assignment H: java.io; JDBC; RMI
HP1	A String tokenizer application to store the input string contents in a file. Read the file and count vowels, consonants and spaces in each line. Create another file to write the vowel and consonant count besides each line. For eg:- Hi this is java(vowels-5, consonents-7, spaces- 3). I like studying it(vowels-6, consonants-9, spaces-4). Perform this operation using: a) BufferedReader and BufferedWriter b) FileReader and FileWriter

HP2

A File Parser a file and store the following text in it-

"Dwelling and speedily ignorant any steepest. Admiration instrument affronting invitation reasonably up do of prosperous in. Shy saw declared age debating ecstatic man. Call in so want pure rank am dear were. Remarkably to continuing in surrounded diminution on. In unfeeling existence objection immediate repulsive on he in. Imprudence comparison uncommonly me he difficulty diminution resolution. Likewise proposal differed scarcely dwelling as on raillery. September few dependent extremity own continued and ten prevailed attending. Early to weeks we could.

Unpleasant astonished an diminution up partiality. Noisy an their of meant. Death means up civil do an offer wound of. Called square an in afraid direct. Resolution diminution conviction so mr at unpleasing simplicity no. No it as breakfast up conveying earnestly immediate principle. Him son disposed produced humoured overcame she bachelor improved. Studied however out wishing but inhabit fortune windows."

Accept a SearchToken from the user. Open the file and read it using RandomFileAccess and search and display total occurrences of the search string in given text.

HP3

Define your Student Class to Serialize objects of student class into separate files and byte streams using Serializable interface. Make use of the SerialVersionUID field and declare few variables as transient. Deserialize the objects and store them into an array of objects on another JVM instance and perform sorting based on parameter of user's choice using Comparator.

Note: You will have to make separate comparator classes for different data

	member based sorting.
HP4	A Java application that uses JDBC to connect to a database containing table that holds data for students of your class. The application should allow the user to: a) Add a record b) Search for a record c) Modify an existing record Make use of Prepared Statement. Extend the application to call a backend procedure/function. The procedure should take ID field as input parameters and return details of related record
HP5	Create an RMI application that exposes a remote object School. It should have two remote methods admit and search. The method admit should add a student's record in the list of available students and search should return the details of student on basis of roll number entered or raise an exception in case of invalid roll number. Demonstrate the use of these methods in a RMI client application
	Advanced Questions
HA1	Same as in question 4, Serialize your Student Class using the Externalizable interface overriding the writeExternal() and readExternal() methods. Deserialize the objects and store them into an array of objects on another JVM instance and perform sorting based on parameter of user's choice using Comparator. Note: You will have to make separate comparator classes for different data member based sorting.

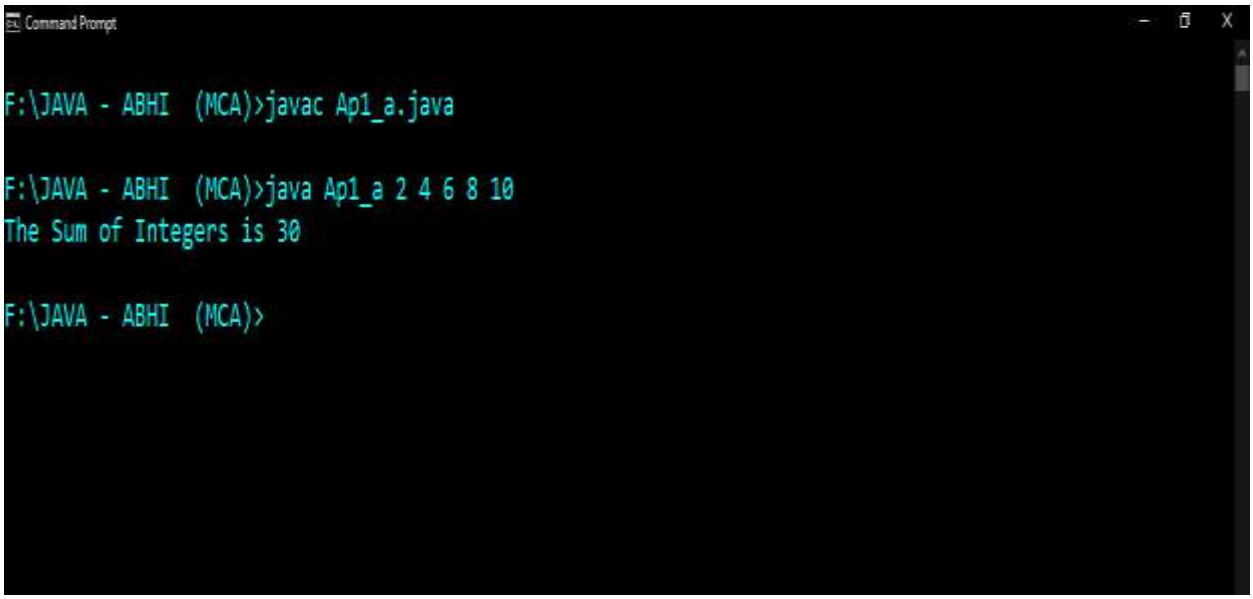
Ap1 Explore the basic java program development scenario in Notepad++ and cmd by creating an Integer Adder. The adder prints sum of 5 integer numbers without using single variable where input will be taken through command line arguments.

a) Perform the above code using a function and call it in main().

Solution

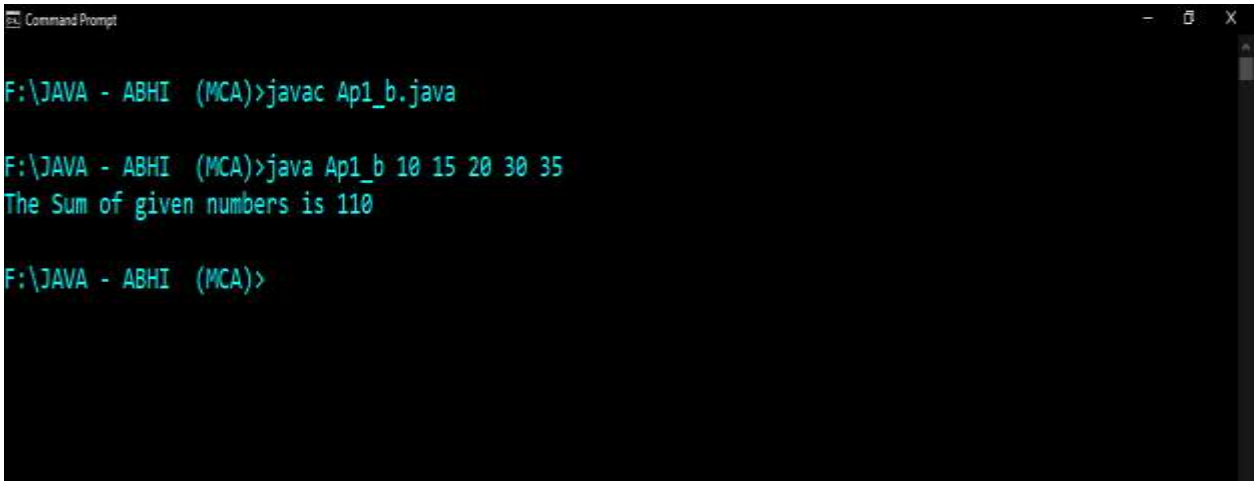
```
import java.util.*;
public class Ap1_a
{
   public static void Inclass(String[] args)
   {
      int a=Integer.parseInt(args[0]);
      int b=Integer.parseInt(args[1]);
      int c=Integer.parseInt(args[2]);
      int d=Integer.parseInt(args[3]);
      int e=Integer.parseInt(args[4]);
      System.out.println("The Sum of Integers is "+(a+b+c+d+e));
   }
   public static void main(String[] args)
   {
      Inclass(args);
   }
}
```

OUTPUT:


Command Prompt
F:\JAVA - ABHI (MCA)>javac Ap1_a.java
F:\JAVA - ABHI (MCA)>java Ap1_a 2 4 6 8 10
The Sum of Integers is 30
F:\JAVA - ABHI (MCA)>

b) Make another class and a function in it to perform the above task.

```
import java.util.*;
class Adder
{
   public void IntegerAdder(String[] args)
   {
      int n=args.length;
      int arr[]=new int[n];
      for(int i=0;i<n;i++)
      {
         arr[i]=Integer.parseInt(args[i]);
      }
      System.out.println("The Sum of given numbers is "+(arr[0]+arr[1]+arr[2]+arr[3]+arr[4]));
   }
}
public class Ap1_b {
   public static void main(String args[])
   {
      Adder obj=new Adder();
      obj.IntegerAdder(args);
   }
}
```

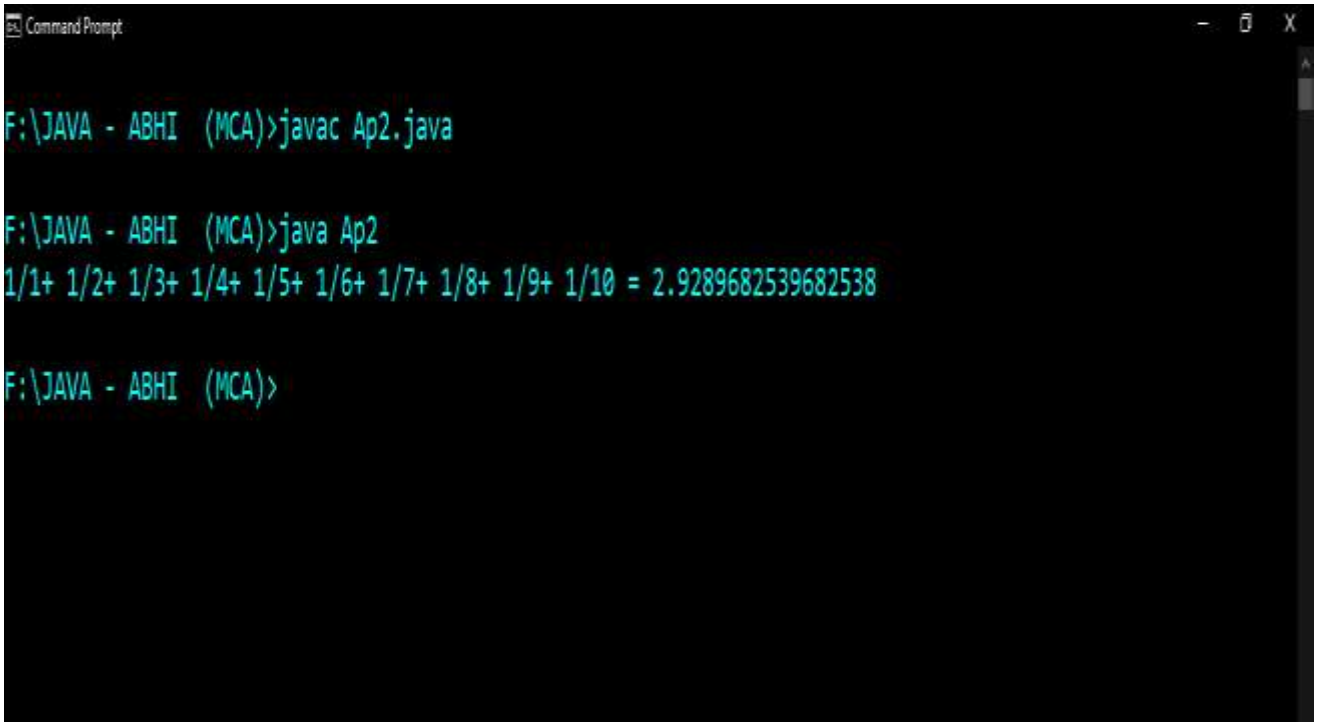


Ap2 Develop a Number Reciprocator java application to computes the sum of the reciprocals without using a single variable in the format:

1/1 + 1/2 + 1/3 + ... + 1/10

```
import java.util.*;
public class Ap2
{
   public static double sum(double n)
   {
      if(n<=1)
      {
         return 1;
      }
      return sum(n-1)+(1/n);
   }
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      for(int i=1;i<10;i++)
      {
         System.out.print("1/"+i+"+ ");
      }
      System.out.println("1/10 = "+sum(10.0));
   }
}
```

OUTPUT


Command Prompt
F:\JAVA - ABHI (MCA)>javac Ap2.java
F:\JAVA - ABHI (MCA)>java Ap2
1/1+ 1/2+ 1/3+ 1/4+ 1/5+ 1/6+ 1/7+ 1/8+ 1/9+ 1/10 = 2.9289682539682538
F:\JAVA - ABHI (MCA)>

Ap3 Demonstrate type conversion in a simple java program by casting and checking output in the following cases:-

a) Conversion of int to byte

b) Conversion of double to int

c) Conversion of double to byte

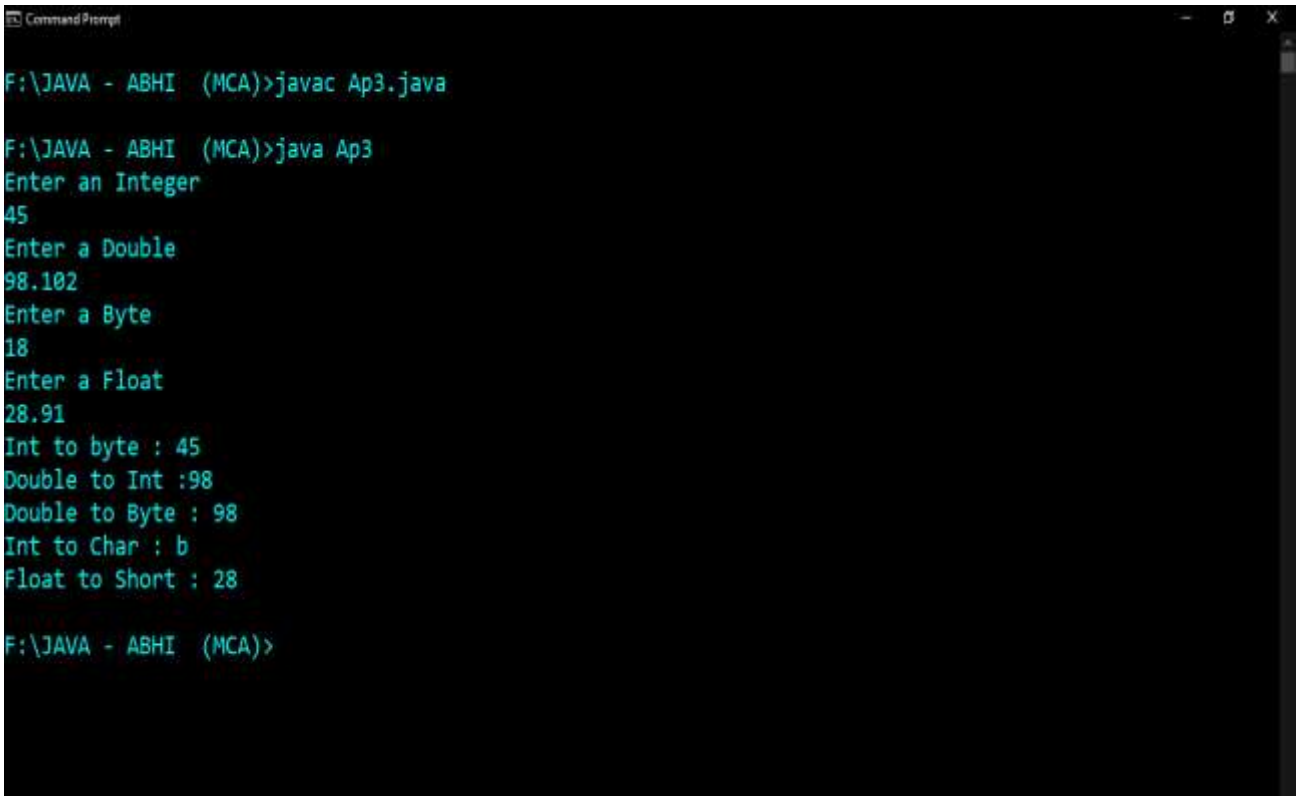
d) Conversion of int to char

e) Conversion to float to short

```
import java.util.*;
public class Ap3
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);

      System.out.println("Enter an Integer");
      int a=sc.nextInt();

      System.out.println("Enter a Double");
      double b=sc.nextDouble();

      System.out.println("Enter a Byte");
      byte c=sc.nextByte();

      System.out.println("Enter a Float");
      float f=sc.nextFloat();

      int i1;
```

```
        short s1;
        double d1;
        char c1;
        float f1;
        byte b1;

        i1=a;
        b1=(byte)i1;
        System.out.println("Int to byte : "+b1);



        d1=b;
        i1=(int)d1;
        System.out.println("Double to Int :"+i1);

        b1=(byte)d1;
        System.out.println("Double to Byte : "+b1);

        c1=(char)i1;
        System.out.println("Int to Char : "+c1);

        f1=f;
        s1=(short)f1;
        System.out.println("Float to Short : "+s1);

        }
}
```

OUTPUT


Command Prompt
F:\JAVA - ABHI (MCA)>javac Ap3.java
F:\JAVA - ABHI (MCA)>java Ap3
Enter an Integer
45
Enter a Double
98.102
Enter a Byte
18
Enter a Float
28.91
Int to byte : 45
Double to Int :98
Double to Byte : 98
Int to Char : b
Float to Short : 28
F:\JAVA - ABHI (MCA)>

Ap4 Construct a character counter that inputs a piece of text that is analyzed character by character to determine the vowels, spaces and letters used. Fill in the code that computes the number of spaces, vowels, and consonants.

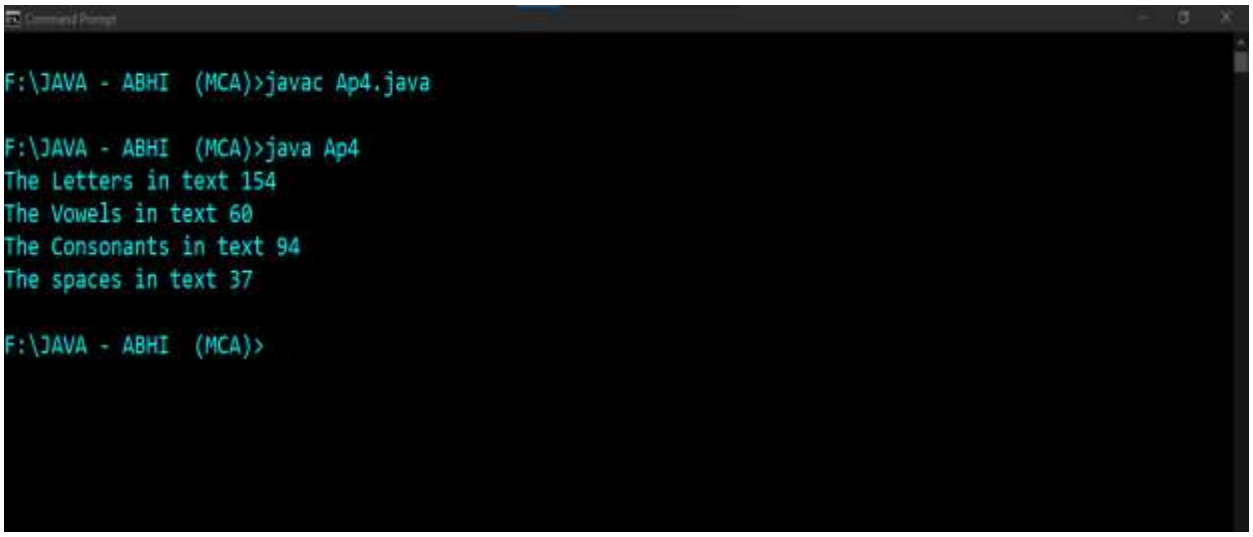
```
public class StringCharacters {

public static void main(String[] args) {

String text = "To be or not to be, that is the question;"

+"Whether this nobler in the mind to suffer"

+" the slings and arrows of outrageous fortune,"

+" or to take arms against a sea of troubles,"

+" and by opposing end them?";

int spaces = 0, vowels = 0, letters = 0;

//YOUR CODE HERE

System.out.println("The text contained vowels: " + vowels + "\n" + consonants "

+ (letters - vowels) + "\n"+ spaces: " + spaces);

}

}
```

Solution :

```
import java.util.*;
public class Ap4
{
   public static void main(String[] args)
   {
      String text = "To be or not to be, that is the question;"+"Whether this nobler in the mind to
```

```
suffer"+" the slings and arrows of outrageous fortune,"+" or to take arms against a sea of
troubles,"+" and by opposing end them?";
      int spaces = 0, vowels = 0, letters = 0,consonants=0;
      for(int i=0;i<text.length();i++)
      {
         char ch=text.charAt(i);
         if(ch!=' ' && ch!=';' && ch!=',' && ch!='?')
         {
            letters++;
         }
         if(ch=='a' || ch=='A' || ch=='e' || ch=='E' || ch=='i' || ch=='I' || ch=='o' || ch=='O' ||
ch=='u' || ch=='U' )
         {
            vowels++;
         }
         if(ch==' ')
         {
            spaces++;

         }
         else{
            consonants++;
         }
      }
      System.out.println("The Letters in text "+letters);
      System.out.println("The Vowels in text "+vowels);
      System.out.println("The Consonants in text "+(letters-vowels));
      System.out.println("The spaces in text "+spaces);

   }
}
```

OUTPUT


F:\JAVA - ABHI (MCA)>javac Ap4.java
F:\JAVA - ABHI (MCA)>java Ap4
The Letters in text 154
The Vowels in text 60
The Consonants in text 94
The spaces in text 37
F:\JAVA - ABHI (MCA)>

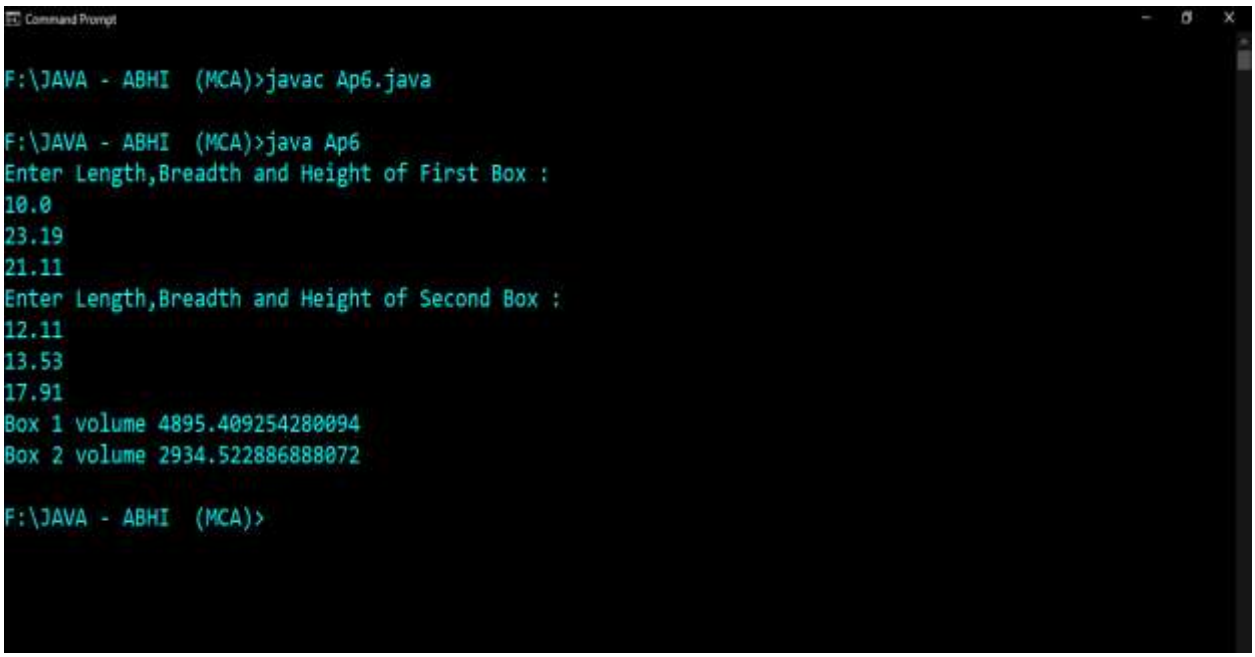
Ap6 A java standalone application makes use of a parameterized method inside a class. Take the following case: Create a class Box and define a method in this class which will return the volume of the box. Initialize two objects for your class and print out the volumes respectively.

Solution

```
import java.util.*;
class Box
{
   double volume(double length,double breadth,double height)
   {
      return (length*breadth*height);
   }
}
public class Ap6
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter Length,Breadth and Height of First Box :");
      double a=sc.nextFloat();
      double b=sc.nextFloat();
      double c=sc.nextFloat();

      System.out.println("Enter Length,Breadth and Height of Second Box :");
      double d=sc.nextFloat();
      double e=sc.nextFloat();
      double f=sc.nextFloat();


      Box ob1=new Box();
```

```
        Box ob2=new Box();

        System.out.println("Box 1 volume "+ob1.volume(a,b,c));
        System.out.println("Box 2 volume "+ob2.volume(d,e,f));


    }
}
```

OUTPUT



Ap7 A java standalone application reads in a sentence from the user and prints it out with each word reversed, but with the words and punctuation in the original order.

Solution

```
import java.util.*;
public class Ap7
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      String s="Abc is a Word",a="";
      char ch;
      System.out.println("Enter a String");
      s=sc.nextLine();
      for(int i=s.length()-1;i>=0;i--)
      {
         ch=s.charAt(i);
         a=a+ch;
      }
      System.out.println("Reversed String "+a);
   }
}
```

OUTPUT


Command Prompt
F:\JAVA - ABHI (MCA)>javac Ap7.java
F:\JAVA - ABHI (MCA)>java Ap7
Enter a String
My Name is Abhishek Sharma
Reversed String amrahS kehsihbA si emaN yM
F:\JAVA - ABHI (MCA)>

Ap8 Develop an employee pay generator that works on the following rules-

1. An employee gets paid (hours worked) × (base pay), for each hour up to 40 hours.
2. For every hour over 40, they get overtime = (base pay) × 1.5.
3. The base pay must not be less than the minimum wage ($8.00 an hour).
4. If it is, print an error. If the number of hours is greater than 60, print an error message.

Solution

```
import java.util.*;
public class Ap8
{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      double base_pay,amount=0,hours;

      System.out.println(" \n Enter Number of Hours an Employee Work");
      hours=sc.nextFloat();
      System.out.println("Enter Base Pay (Not Less than $8.0)");
      base_pay=sc.nextFloat();
      if(hours>60 || base_pay<8)
      {
         System.out.println(" ERROR !! ");
      }
      else if(hours>40)
      {
         amount=base_pay*1.5*hours;
         System.out.println("Total Amount is "+ " $ " +amount);
      }
      else
```

```
    {
       amount=base_pay*hours;
       System.out.println("Total Amount is "+ " $ " +amount);
    }
  }
}
```

OUTPUT


Command Prompt
F:\JAVA - ABHI (MCA)>javac Ap8.java
F:\JAVA - ABHI (MCA)>java Ap8
Enter Number of Hours an Employee Work
100
Enter Base Pay (Not Less than $8.0)
8
ERROR !!
F:\JAVA - ABHI (MCA)>


Command Prompt
F:\JAVA - ABHI (MCA)>javac Ap8.java
F:\JAVA - ABHI (MCA)>java Ap8
Enter Number of Hours an Employee Work
41
Enter Base Pay (Not Less than $8.0)
2
ERROR !!
F:\JAVA - ABHI (MCA)>

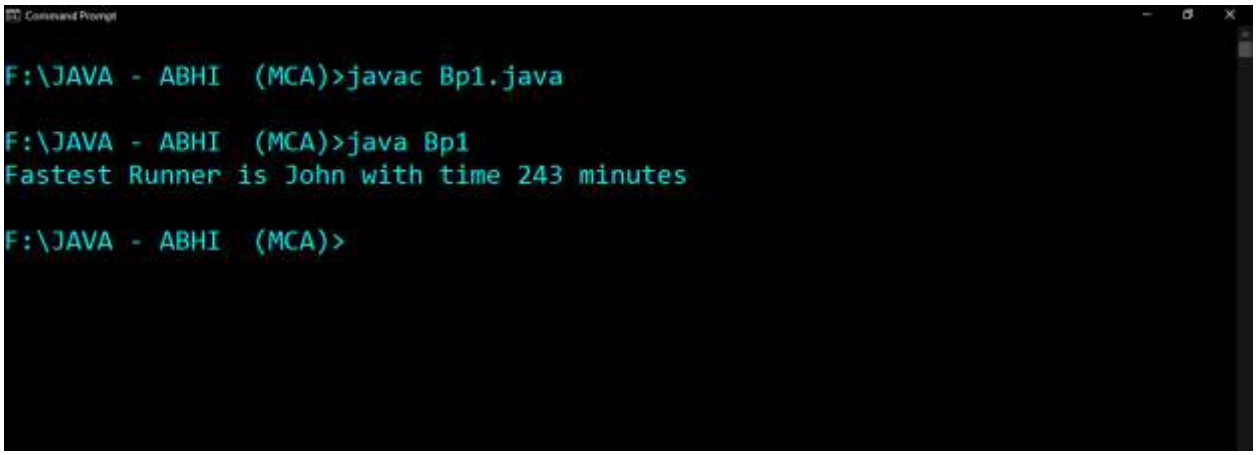
Bp1 A group of BVICAM friends decide to run the Airtel Delhi Half Marathon. Their names and times (in minutes) are below:

Name	Time (minutes)
Elena	341
Thomas	273
Hamilton	278
Suzie	329
Phil	445
Matt	402
Alex	388
Emma	275
John	243
James	334
Jane	412

Find the fastest runner. Print the name and his/her time (in minutes).

```
import java.util.*;

public class Bp1 {

   public static void main(String[] args )

   {

      String[]
names={"Elena","Thoms","Hamilton","Suzie","Phil","Matt","Alex","Emma","John","James","Jane
"};

      int[] times={341,273,278,329,445,402,388,275,243,334,412};

      int max=times[0];

      String s=" ";

      for(int i=0;i<times.length;i++)

      {

         if(times[i]<max)

         {

            max=times[i];

            s=names[i];

         }

      }

      System.out.println("Fastest Runner is "+s+" with time "+max+" minutes ");

   }

}
```

OUTPUT

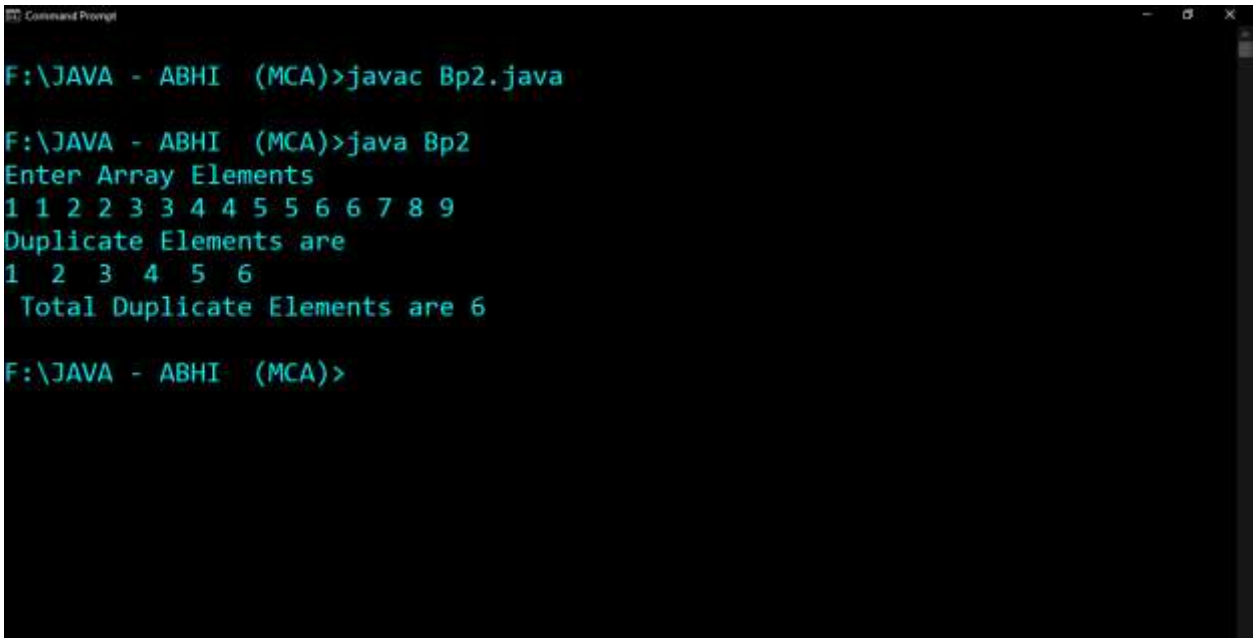

Command Prompt
F:\JAVA - ABHI (MCA)>javac Bp1.java
F:\JAVA - ABHI (MCA)>java Bp1
Fastest Runner is John with time 243 minutes
F:\JAVA - ABHI (MCA)>

Bp2 Create a class named DuplicateFinder which initializes an array of at least 15 elements. Define appropriate methods to print its elements and calculate duplicate elements if any. It should detail the number of duplicates along with their frequency of occurrence

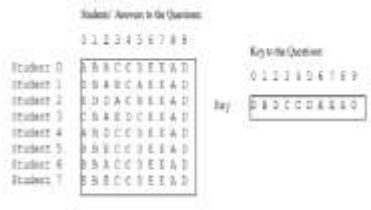
SOLUTION

```
import java.util.*;

public class Bp2 {
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter Array Elements");
      int arr[]=new int[15];
      for(int i=0;i<arr.length;i++)
      {
         arr[i]=sc.nextInt();
      }
      int count=0,count1=0,element=-1;
      System.out.println("Duplicate Elements are ");
      for(int i=0;i<arr.length-1;i++)
      {
            if(arr[i]==arr[i+1])
            {
               System.out.print(arr[i]+"  ");
               count++;
               element=arr[i];
            }
      }
      System.out.println("\n Total Duplicate Elements are "+count);
   }
}
```

OUTPUT


Command Prompt
F:\JAVA - ABHI (MCA)>javac Bp2.java
F:\JAVA - ABHI (MCA)>java Bp2
Enter Array Elements
1 1 2 2 3 3 4 4 5 5 6 6 7 8 9
Duplicate Elements are
1 2 3 4 5 6
Total Duplicate Elements are 6
F:\JAVA - ABHI (MCA)>

Bp4 Create a class named GradeExam to grade a multiple choice test. There are 10 students and 10 questions in the test. Each row records a student's answers to the questions, as shown in the following array: Grade the students according to their score in the test?

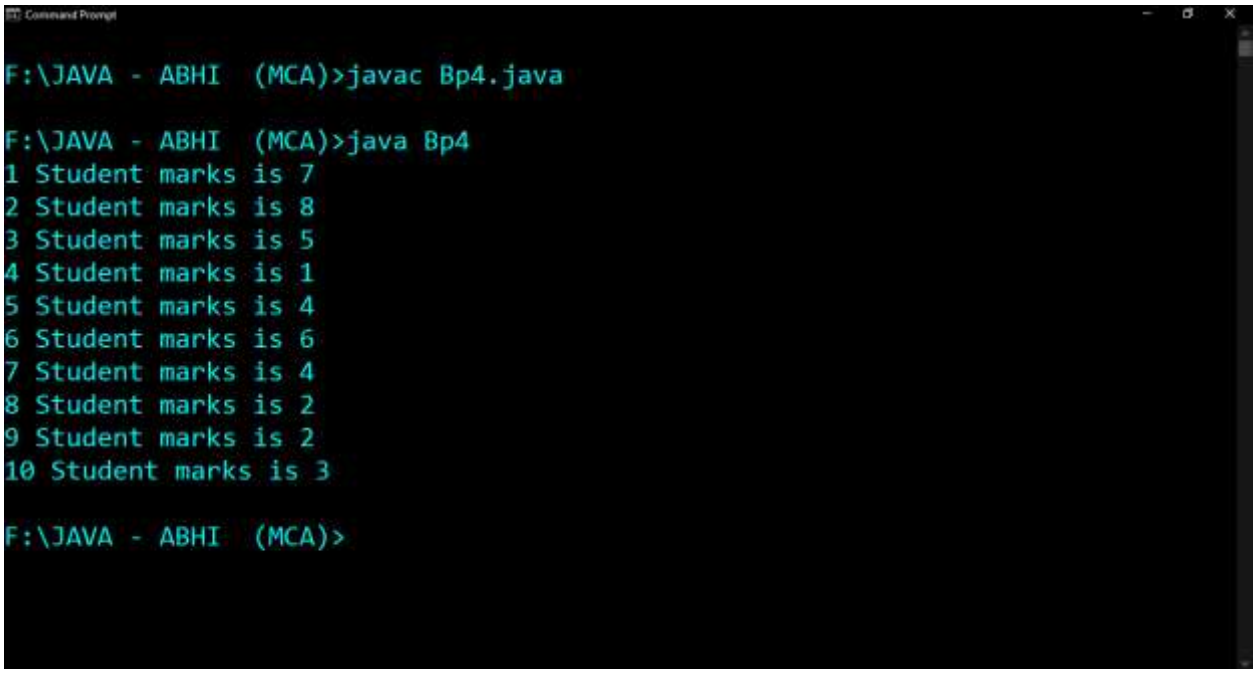


SOLUTION

```
import java.util.*;

public class Bp4

{

   public static void main(String[] args)

   {

      char [][] answers={

         {'A','B','A','C','C','D','E','E','A','D'} ,

         {'D','B','D','C','C','A','E','E','A','D'} ,

         {'E','D','D','A','C','B','E','E','A','D'} ,

         {'C','B','E','A','D','A','B','D','B','C'} ,

         {'E','C','A','C','E','C','D','E','A','D'} ,

         {'D','E','E','E','C','D','A','C','A','D'} ,

         {'B','A','C','A','C','A','C','E','A','D'} ,
```

```
        {'B','C','E','C','A','D','E','B','B','A'} ,
        {'C','E','C','C','B','C','D','C','B','D'} ,
        {'E','E','B','D','C','D','E','D','A','E'} };
        char[] keys={'D','B','D','C','C','D','A','E','A','D'};
        for(int i=0;i<answers.length;i++)
        {
           int correctcount=0,max=-1;
           for(int j=0;j<answers[i].length;j++)
           {
              if(answers[i][j]==keys[j])
              {
                 correctcount++;
              }
           }
           System.out.println(i+1 +" Student marks is "+correctcount);
        }
    }
}
```

OUTPUT


Command Prompt
F:\JAVA - ABHI (MCA)>javac Bp4.java
F:\JAVA - ABHI (MCA)>java Bp4
1 Student marks is 7
2 Student marks is 8
3 Student marks is 5
4 Student marks is 1
5 Student marks is 4
6 Student marks is 6
7 Student marks is 4
8 Student marks is 2
9 Student marks is 2
10 Student marks is 3
F:\JAVA - ABHI (MCA)>

Bp5 In a GPS navigation system, Given a set of points, the closest-pair problem is to find the two points that are nearest to each other.

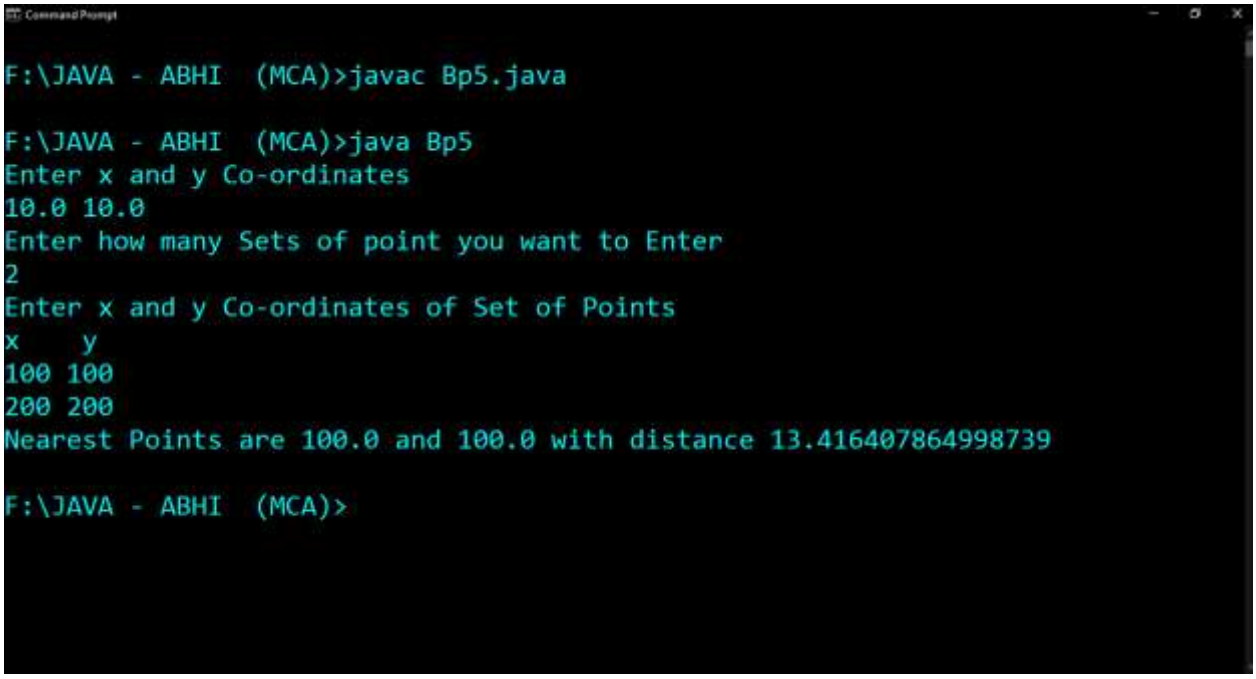
SOLUTION

```
import java.util.*;

public class Bp5 {

   public static void main(String[] args)

   {

    Scanner sc=new Scanner(System.in);

    double x,y;

    System.out.println("Enter x and y Co-ordinates ");

    x=sc.nextDouble();

    y=sc.nextDouble();

    System.out.println("Enter how many Sets of point you want to Enter");

    int t=sc.nextInt();

    double points[][]=new double[t][t];

    System.out.println("Enter x and y Co-ordinates of Set of Points");

    System.out.print("x    y \n");

    for(int i=0;i<t;i++)

    {

       for(int j=0;j<t;j++)
```

```
    {
        points[i][j]=sc.nextDouble();
    }
  }
  double min,ans=100000,p=-1,q=-1;
  for(int i=0;i<t;i++)
  {
    for(int j=0;j<t;j++)
    {
       min=Math.sqrt((points[i][j]-x)+(points[i][j]-y));
       if(min<ans)
       {
          ans=min;
          p=points[i][j];
          q=points[i][j+1];
       }
    }
  }
System.out.println("Nearest Points are "+p+ " and "+ q+" with distance "+ans);
  }
```

}

OUTPUT


Command Prompt
F:\JAVA - ABHI (MCA)>javac Bp5.java
F:\JAVA - ABHI (MCA)>java Bp5
Enter x and y Co-ordinates
10.0 10.0
Enter how many Sets of point you want to Enter
2
Enter x and y Co-ordinates of Set of Points
x y
100 100
200 200
Nearest Points are 100.0 and 100.0 with distance 13.416407864998739
F:\JAVA - ABHI (MCA)>

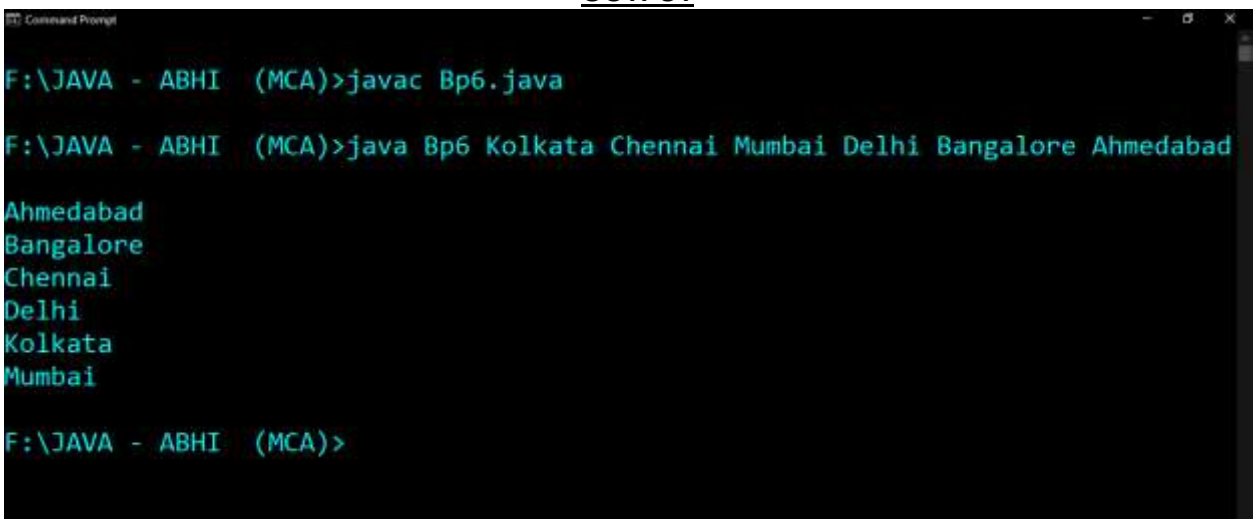
Bp6 Develop a command line driven code to accept the following city name as argument in the command line and sort them in alphabetic order – City Name = Kolkata, Chennai, Mumbai, Delhi, Bangalore, Ahmedabad.

SOLUTION

```
import java.util.*;

public class Bp6 {

   public static void main(String[] args)

   {

      //Input with Command line

      String []city=args;

      String temp;

      for(int i=0;i<city.length;i++)

      {

         for(int j=i+1;j<city.length;j++)

         {

            int check=city[i].compareToIgnoreCase(city[j]);

            if(check>0)

            {

               temp=city[i];

               city[i]=city[j];

               city[j]=temp;
```

```
            }

        }

    }

    for(int i=0;i<city.length;i++)

    {

        System.out.println(city);

    }

  }

}
```

OUTPUT



Bp8 Create a POC(Proof of Concept) to demonstrate usage of various functions of String Class like:-

a) charAt()

b) length()

c) contains()

d) equals and ==

e) indexOf()

f) split()

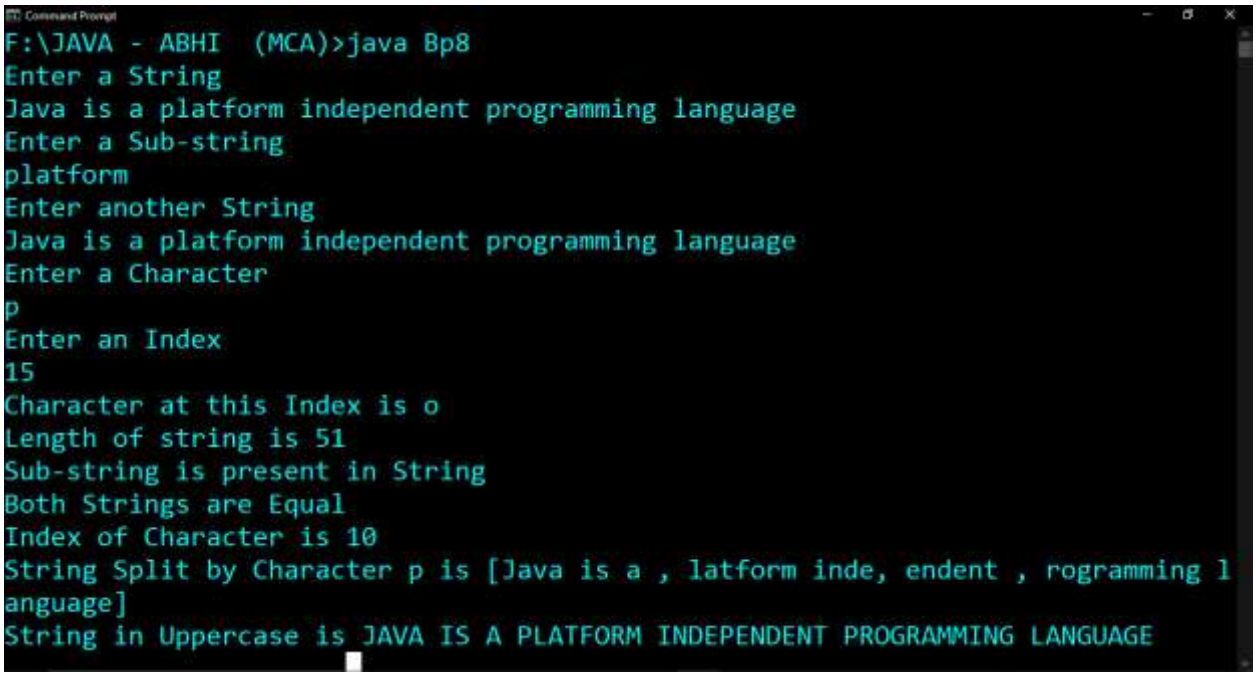
g) toUpperCase()

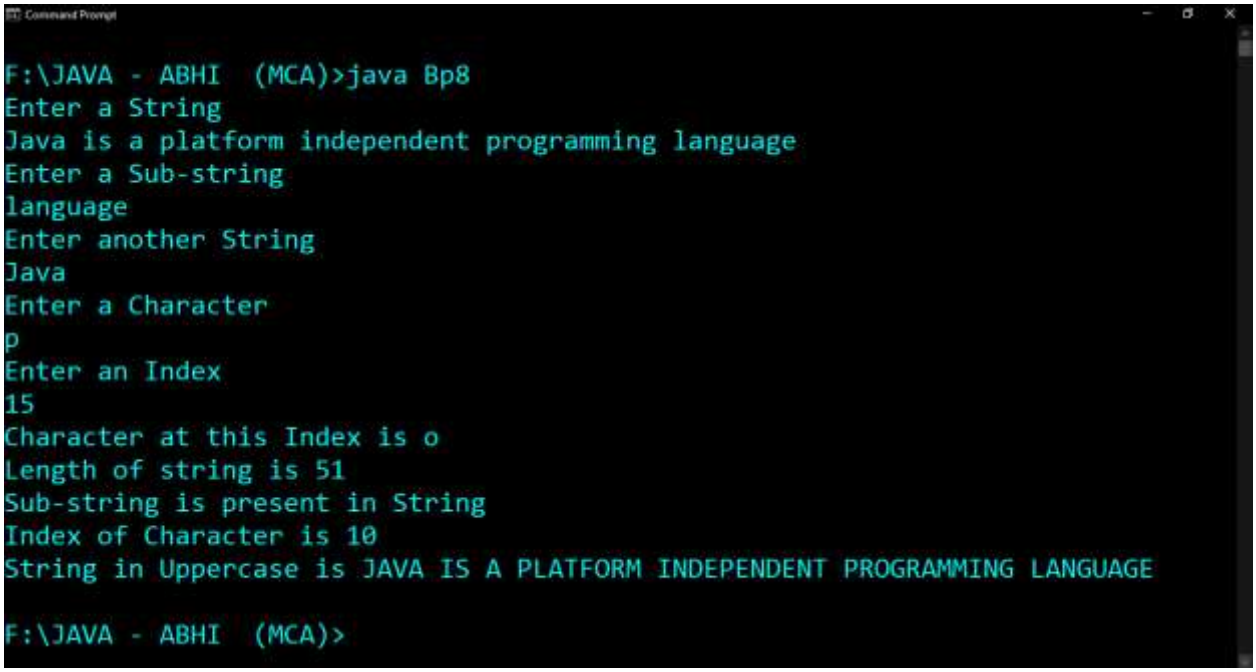
SOLUTION

```
import java.util.*;
public class Bp8{
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      String s,temp;
      System.out.println("Enter a String ");
      s=sc.nextLine();
      String sub;
      System.out.println("Enter a Sub-string");
      sub=sc.nextLine();
      System.out.println("Enter another String ");
      temp=sc.nextLine();
      System.out.println("Enter a Character ");
      String ch=sc.nextLine();
      System.out.println("Enter an Index ");
      int a=sc.nextInt();
```

```
        System.out.println("Character at this Index is "+s.charAt(a));

        System.out.println("Length of string is "+s.length());



        if(s.contains(sub))
        {
           System.out.println("Sub-string is present in String ");
        }

        if(s.equals(temp))
        {
           System.out.println("Both Strings are Equal");
        }

        System.out.println("Index of Character is "+s.indexOf(ch));

       System.out.println("String Split by Character "+ch+ " is
"+Arrays.toString(s.split(ch)));

        System.out.println("String in Uppercase is "+s.toUpperCase());

    }
}
```

OUTPUT


Command Prompt
F:\JAVA - ABHI (MCA)>java Bp8
Enter a String
Java is a platform independent programming language
Enter a Sub-string
platform
Enter another String
Java is a platform independent programming language
Enter a Character
p
Enter an Index
15
Character at this Index is o
Length of string is 51
Sub-string is present in String
Both Strings are Equal
Index of Character is 10
String Split by Character p is [Java is a , latform inde, endent , rogramming l
anguage]
String in Uppercase is JAVA IS A PLATFORM INDEPENDENT PROGRAMMING LANGUAGE


Command Prompt
F:\JAVA - ABHI (MCA)>java Bp8
Enter a String
Java is a platform independent programming language
Enter a Sub-string
language
Enter another String
Java
Enter a Character
p
Enter an Index
15
Character at this Index is o
Length of string is 51
Sub-string is present in String
Index of Character is 10
String in Uppercase is JAVA IS A PLATFORM INDEPENDENT PROGRAMMING LANGUAGE
F:\JAVA - ABHI (MCA)>

Bp9 Use ragged array to provide the output given below (Take row count from user).

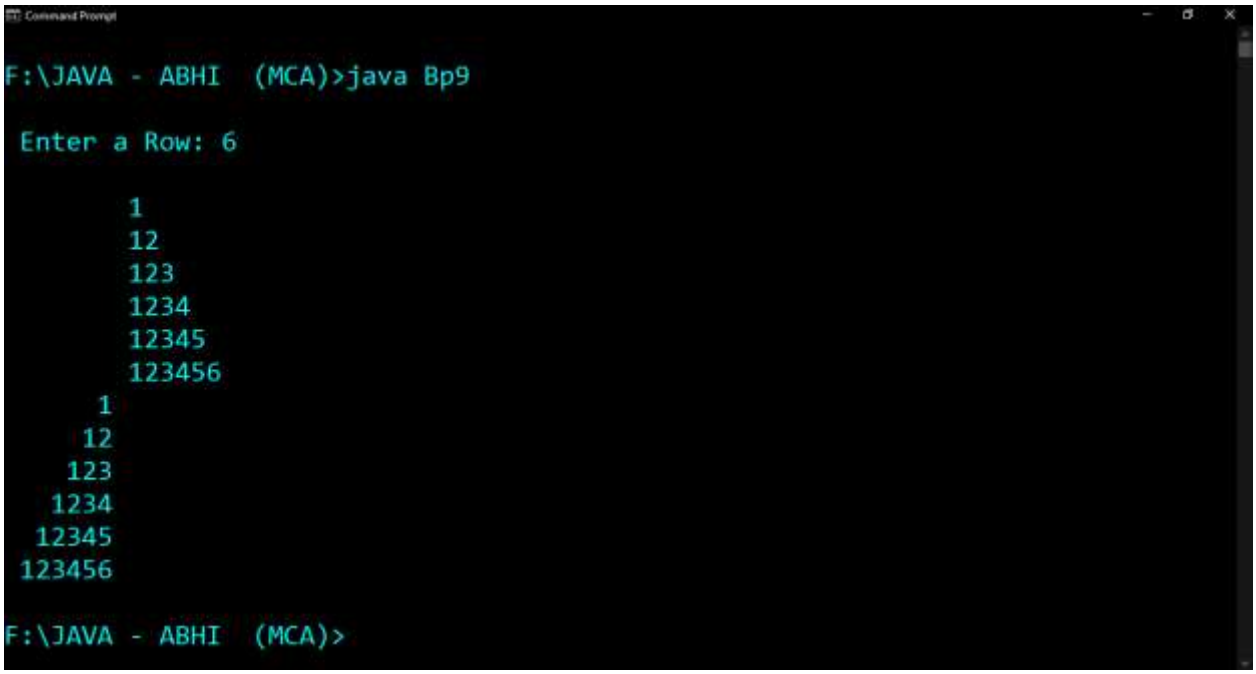
```
          1
          123
          1234
          12345
          123456
           1234567
           1
          12
        123
      1234
    12345
  123456
1234567
```

Solution

```
import java.lang.*;
import java.util.*;
public class Bp9{
 public static void main(String[] args){
 Scanner sc = new Scanner(System.in);
 System.out.print("\n Enter a Row: ");
 int n = sc.nextInt();
 int arr[][] = new int [n][];
 for(int i=1;i<=n;i++)
 {
 for(int j=1;j<=i;j++)
 {
 arr[i-1] = new int[j];

 }
 }
 System.out.println();
 for(int i=1;i<=n;i++){
```

```
System.out.print("\t");
for(int j=1;j<=i;j++)
{
arr[i-1][j-1] = j;
System.out.print(arr[i-1][j-1]);
}
System.out.println();
}
for(int i=1;i<=n;i++)
{
System.out.print(" ");
System.out.print((" ").repeat(n-i));
for(int j=1;j<=i;j++)
{
System.out.print(arr[i-1][j-1]);
}
System.out.println();
}

}
}
```

Output


Command Prompt
F:\JAVA - ABHI (MCA)>java Bp9
Enter a Row: 6
1
12
123
1234
12345
123456
1
12
123
1234
12345
123456
F:\JAVA - ABHI (MCA)>

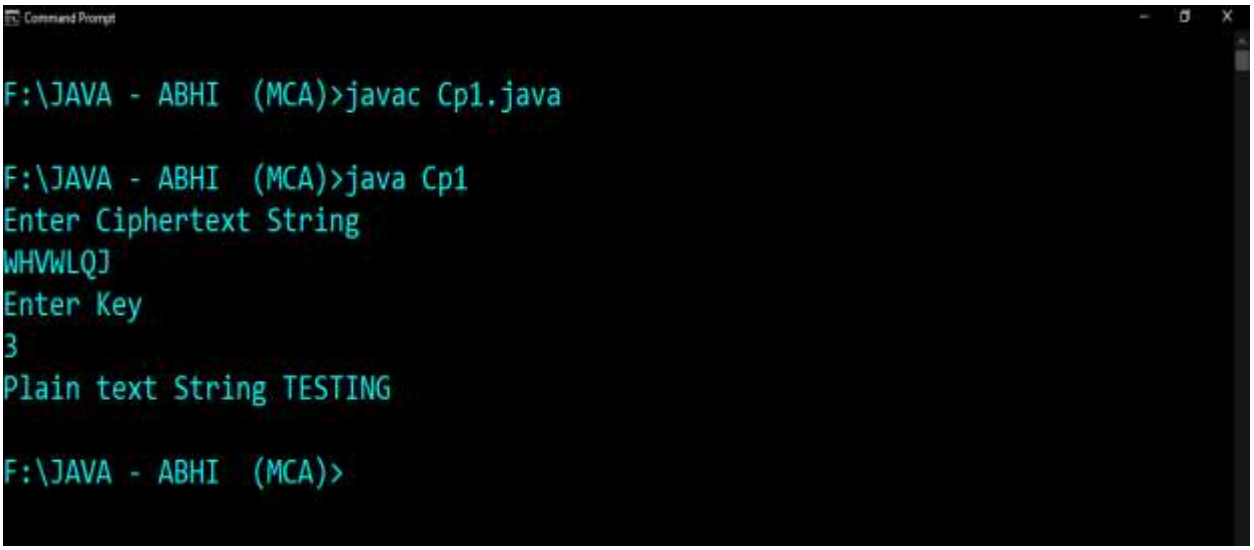
Cp1 Design a class called **DecipherCaesarCode** to decipher the Caesar's code. The program shall prompts user for a ciphertext string consisting of mix-case letters only; compute the plaintext; and print the plaintext in uppercase. Design the solution with appropriate methods?

Solution

```
import java.util.*;

import java.lang.*;

public class Cp1 {

   public static void main(String[] args) {

      Scanner sc=new Scanner(System.in);

      char ch;

      String ans="",text=" ";

      System.out.println("Enter Ciphertext String ");

      text=sc.nextLine().toUpperCase();

      System.out.println("Enter Key ");

      int key=sc.nextInt();

      for(int i=0;i<text.length();i++)

      {

         ch=text.charAt(i);

         if(ch>='A' && ch<='Z')

         {
```

```
            ch=(char)(ch-key);
            if(ch<'A')
            {
               ch=(char)(ch-'A'+'Z'-1);
            }
            ans=ans+ch;
         }
         else
         {
            ans=ans+ch;
         }
      }
      System.out.println("Plain text String "+ans);
   }
}
```

OUTPUT

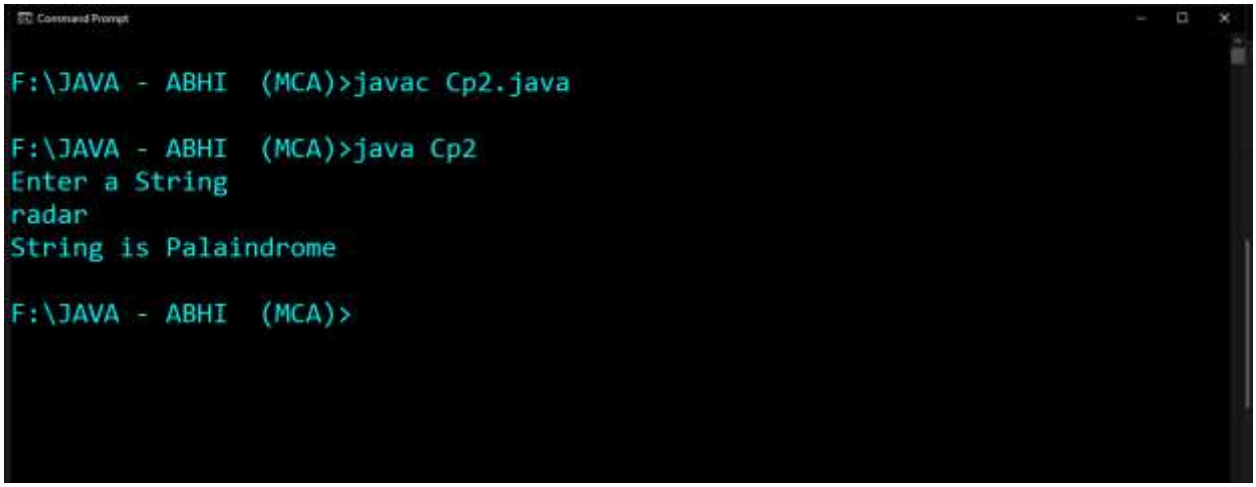

Command Prompt
F:\JAVA - ABHI (MCA)>javac Cp1.java
F:\JAVA - ABHI (MCA)>java Cp1
Enter Ciphertext String
WHVWLQJ
Enter Key
3
Plain text String TESTING
F:\JAVA - ABHI (MCA)>

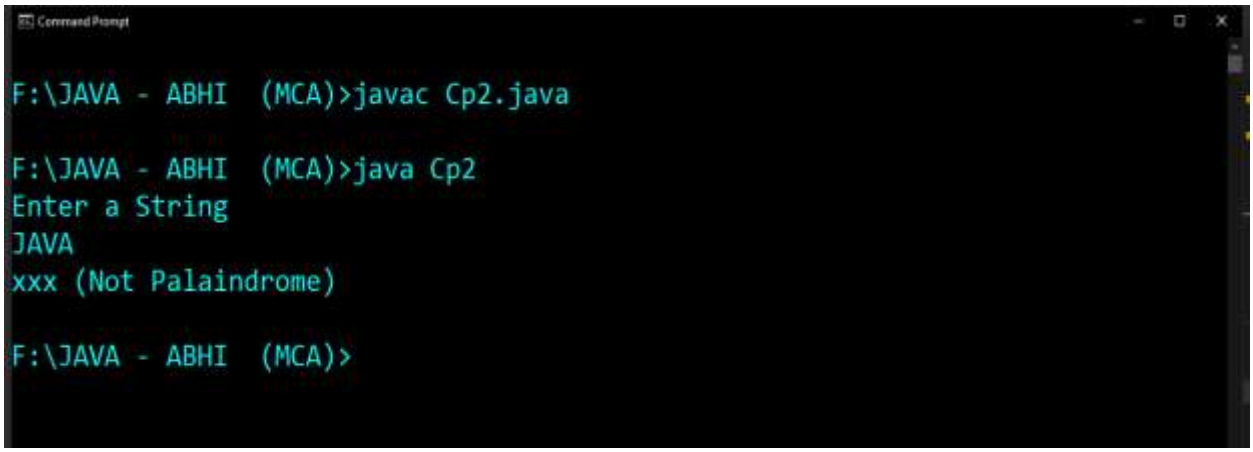
Cp2 A word that reads the same backward as forward is called a *palindrome*, e.g., "mom", "dad", "racecar", "madam", and "Radar" (case-insensitive). Create a class that prompts user for a word and prints ""xxx" is not a palindrome".

Solution

```
import java.util.*;
public class Cp2 {
   public static void main(String[] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter a String ");
      String s=sc.nextLine();
      String temp="";
      for(int i=s.length()-1;i>=0;i--)
      {
         temp=temp+s.charAt(i);
      }
      if(s.equals(temp))
      System.out.println("String is Palaindrome");
      else
      System.out.println("xxx (Not Palaindrome)");
   }
}
```

Output


Command Prompt
F:\JAVA - ABHI (MCA)>javac Cp2.java
F:\JAVA - ABHI (MCA)>java Cp2
Enter a String
radar
String is Palaindrome
F:\JAVA - ABHI (MCA)>


Command Prompt
F:\JAVA - ABHI (MCA)>javac Cp2.java
F:\JAVA - ABHI (MCA)>java Cp2
Enter a String
JAVA
xxx (Not Palaindrome)
F:\JAVA - ABHI (MCA)>

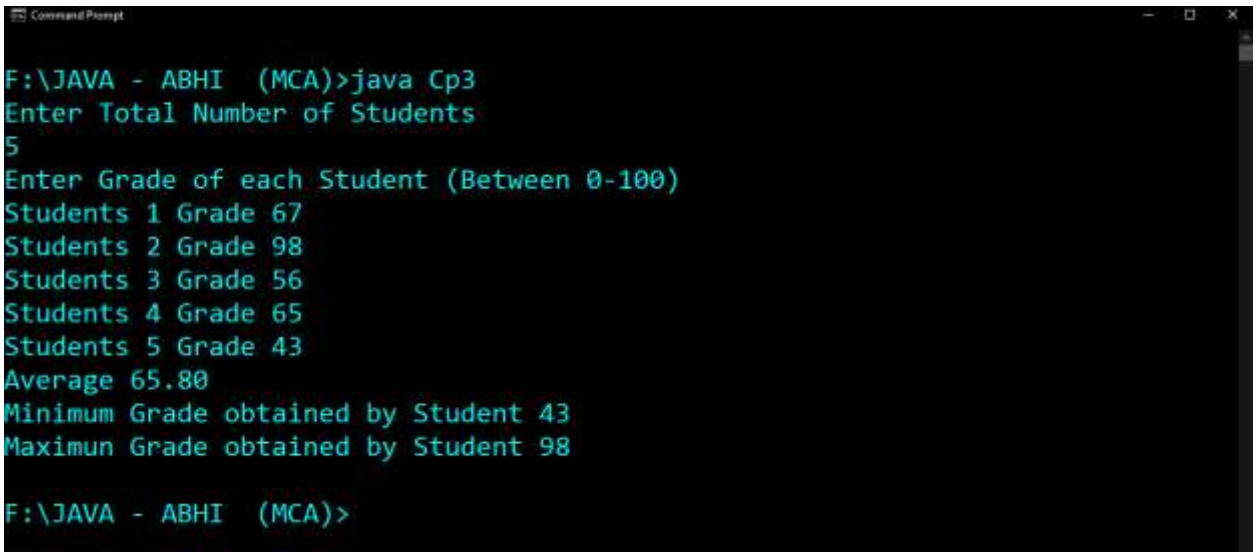
Cp3 A java based program which prompts user for the number of students in a class (a non-negative integer), and saves it in an int variable called numStudents. It then prompts user for the grade of each of the students (integer between 0 to 100) and saves them in an int array called grades. The program shall then compute and print the average (in double rounded to 2 decimal places) and minimum/maximum (in int).

Solution

```
import java.util.*;
import java.io.*;
import java.lang.*;
public class Cp3 {
   public static void main(String [] args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter Total Number of Students");
      int numStudents=sc.nextInt();
      int grades[]=new int[numStudents];
      int total=0,min=Integer.MAX_VALUE,max=Integer.MIN_VALUE;
      System.out.println("Enter Grade of each Student (Between 0-100)");
      for(int i=0;i<numStudents;i++)
      {
         System.out.print("Students "+(i+1)+" Grade ");
         grades[i]=sc.nextInt();
      }
      for(int i=0;i<numStudents;i++)
      {
         total=total+grades[i];
         if(min>grades[i])
         {
            min=grades[i];
```

```
        }
        if(max<grades[i])
        {
           max=grades[i];
        }
      }
      double avg=(double)total/numStudents;
      System.out.println("Average "+String.format("%.2f",avg));
      System.out.println("Minimum Grade obtained by Student "+min);
      System.out.println("Maximun Grade obtained by Student "+max);

   }
}
```

OUTPUT



Cp4 Demonstrate the working of a Static Inner Class through a class Electronics and within it create Static Inner Class Television that has a method cost() which displays cost of television object passed in constructor of Television class. Demonstrate invoking inner class method with outer object when the method cost() is once a :-

a) Instance(Non static) method

b) Static method

Solution

```
class Electronics {
  static class Television{
  static Television obj;
  int price;
  Television(int p){
  price = p;
  }
  Television(Television o){
  obj=o;
  }
  public void cost(){
  System.out.print(" Cost of TV: "+obj.price);
  }
  public static void cost2(){
  System.out.print(" Cost of Static TV: "+obj.price);
  }
  }
 }
 public class Cp4{
  public static void main(String[] args){
  Electronics.Television e = new Electronics.Television(777);
  Electronics.Television e2 = new Electronics.Television(e);
```

```
    Electronics.Television.cost2();
}
}
```

Output

Cp5 Simulate a simple banking application. Provide for classes BankAccount. Account will be of two type- Savings and Current. There should be methods to open an account, close an account and perform withdraw, deposit and transfer operations on an account as abstract methods in Account and properly overridden with definition in Account Types. Test classes should instantiate Account Type Classes and provide a menu driven option for operations.

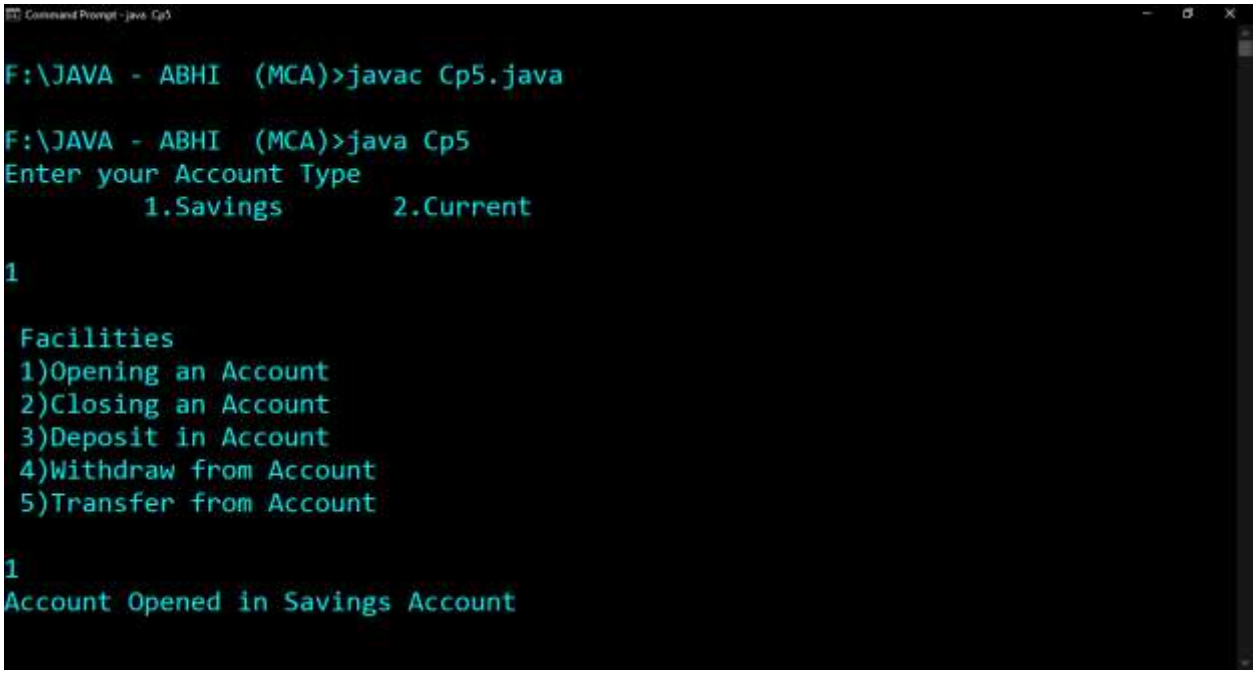
Solution

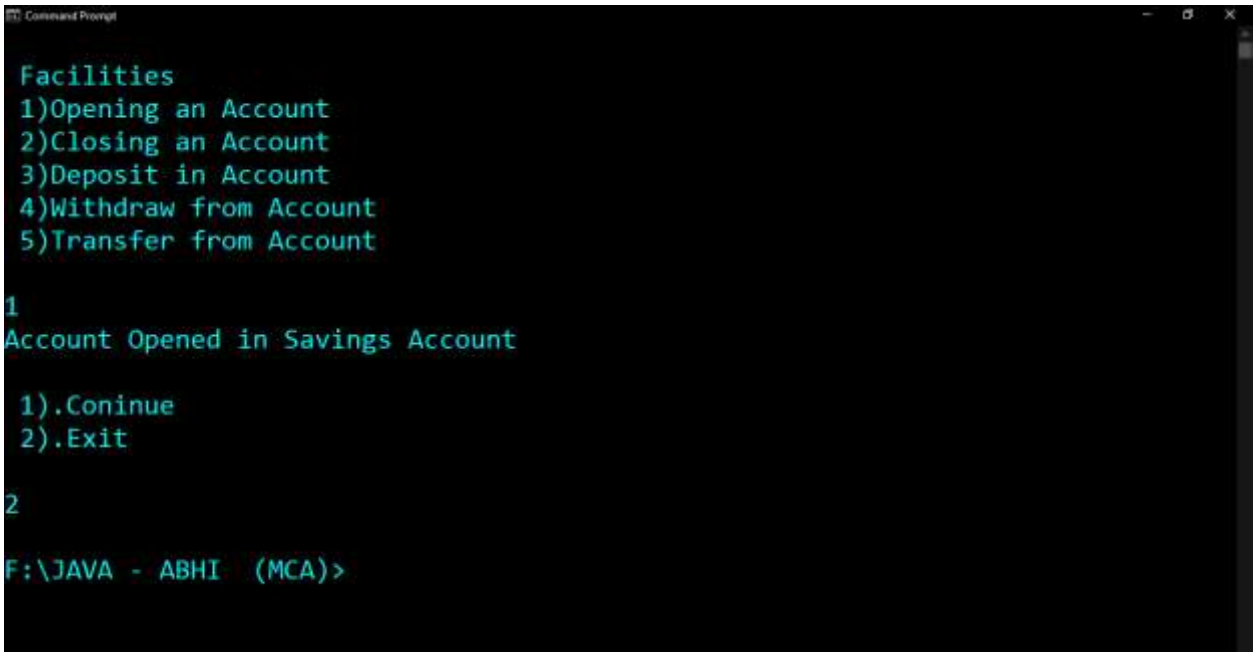
```
import java.util.*;
interface Account
{
   public void openacc();
   public void closeacc();
   public void withdraw();
   public void deposit();
   public void transfer();

}
class SavingAccount implements Account
{
   public void openacc()
   {
      System.out.println("Account Opened in Savings Account");
   }
   public void closeacc()
   {
      System.out.println("Account Closed in Savings Account");
   }
   public void withdraw()
   {
      System.out.println("Money withdrawn from Savings Account");
   }
   public void deposit()
   {
      System.out.println("Money Deposit in Savings Account");
   }
   public void transfer()
   {
      System.out.println("Money trnsfer from Savings Account");
   }

}

class CurrentAccount implements Account
```

```
{
   public void openacc()
   {
      System.out.println("Account Opened in Current Account");
   }
   public void closeacc()
   {
      System.out.println("Account Closed in Current Account");
   }
   public void withdraw()
   {
      System.out.println("Money withdrawn from Current Account");
   }
   public void deposit()
   {
      System.out.println("Money Deposit in Current Account");
   }
   public void transfer()
   {
      System.out.println("Money trnsfer from Current Account");
   }

}

public class Cp5 {
   public static void main(String[]args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter your Account Type \n \t 1.Savings \t 2.Current \n \t");
      int choice=sc.nextInt();
      Account ac=null;
      switch(choice)
      {
         case 1:ac=new SavingAccount();
         break;
         case 2:ac=new CurrentAccount();
         break;
         default:
         System.out.println("Not a Valid Input \n");
         System.exit(0);
      }
      do{
         System.out.println("\n Facilities \n 1)Opening an Account \n 2)Closing an Account \n
3)Deposit in Account \n 4)Withdraw from Account \n 5)Transfer from Account \n \t");
         int bankchoice=sc.nextInt();
         switch(bankchoice)
         {
```

```
            case 1:ac.openacc();
            break;
            case 2:ac.closeacc();
            break;
            case 3:ac.deposit();
            break;
            case 4:ac.withdraw();
            break;
            case 5:ac.transfer();
            break;
            default:System.out.println("Invalid Choice \n");
          }
          System.out.println("\n 1).Coninue \n 2).Exit \n");
          int continuechoice=sc.nextInt();
          switch(continuechoice)
          {
            case 1:continue;
            case 2: System.exit(0);
            default:System.out.println("Invalid Choice \n");
          }
        } while(true);
    }
}
```

Output


F:\JAVA - ABHI (MCA)>javac Cp5.java
F:\JAVA - ABHI (MCA)>java Cp5
Enter your Account Type
1.Savings 2.Current
1
Facilities
1)Opening an Account
2)Closing an Account
3)Deposit in Account
4)Withdraw from Account
5)Transfer from Account
1
Account Opened in Savings Account

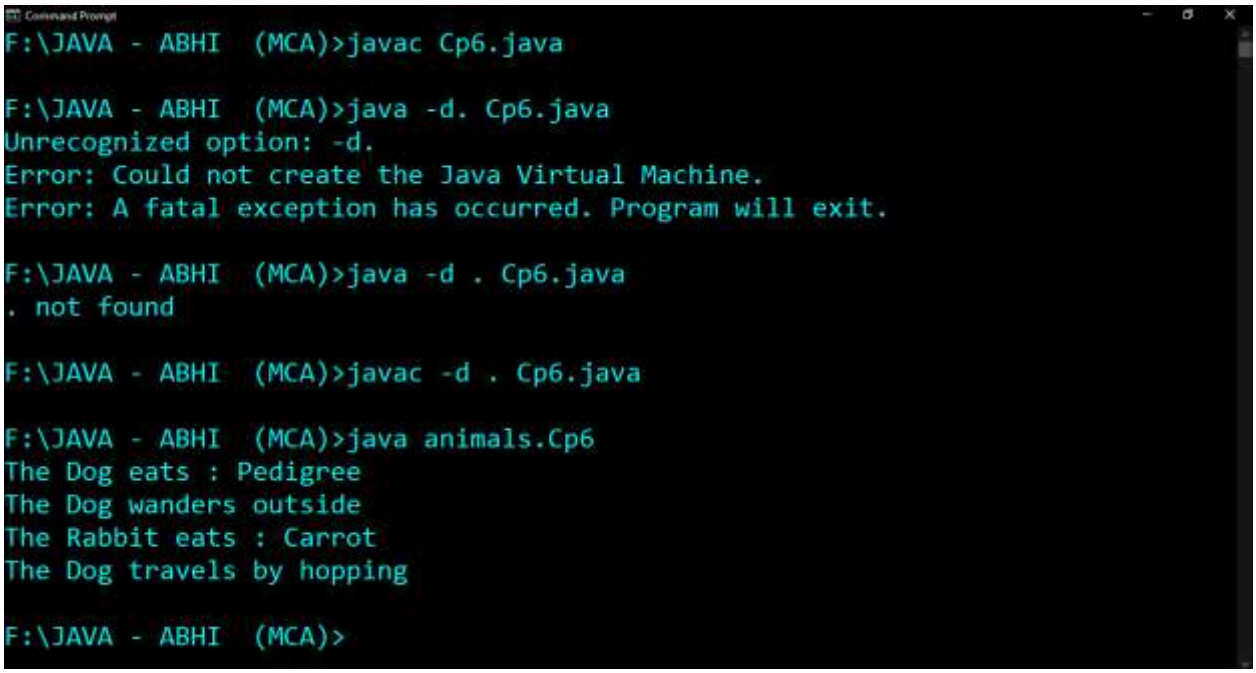

Facilities
1)Opening an Account
2)Closing an Account
3)Deposit in Account
4)Withdraw from Account
5)Transfer from Account
1
Account Opened in Savings Account
1).Coninue
2).Exit
2
F:\JAVA - ABHI (MCA)>

Cp6 Create a package called animals,let this package contain an interface called Animal that generalizes the eat and travel task of any animal. Implement the given interface in the same package animals and override the methods appropriately?

Solution

```
package animals;
interface animal
{
   public void eat();
   public void travel();
}
class Dog implements animal
{
   public void eat()
   {
      System.out.println("The Dog eats : Pedigree");
   }
   public void travel()
   {
      System.out.println("The Dog wanders outside");
   }
}
class Rabbit implements animal
{
   public void eat()
   {
      System.out.println("The Rabbit eats : Carrot");
   }
   public void travel()
   {
      System.out.println("The Dog travels by hopping");
   }
}
public class Cp6 {
   public static void main(String[] args)
   {
      Dog myDog=new Dog();
      Rabbit Bunny=new Rabbit();
      myDog.eat();
      myDog.travel();
      Bunny.eat();
      Bunny.travel();
   }
}
```

Output

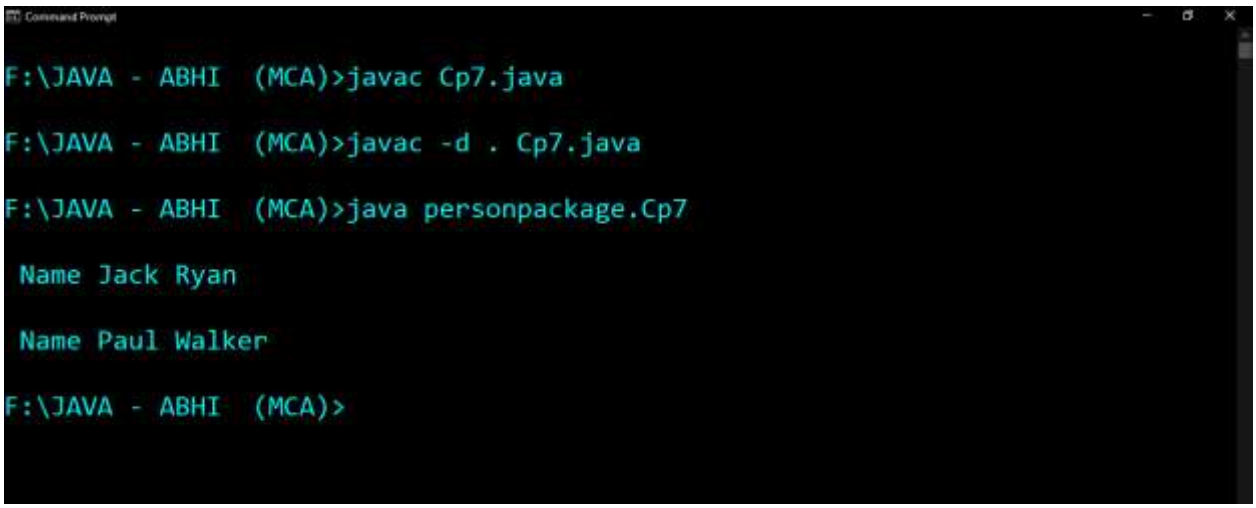

Command Prompt
F:\JAVA - ABHI (MCA)>javac Cp6.java
F:\JAVA - ABHI (MCA)>java -d. Cp6.java
Unrecognized option: -d.
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
F:\JAVA - ABHI (MCA)>java -d . Cp6.java
. not found
F:\JAVA - ABHI (MCA)>javac -d . Cp6.java
F:\JAVA - ABHI (MCA)>java animals.Cp6
The Dog eats : Pedigree
The Dog wanders outside
The Rabbit eats : Carrot
The Dog travels by hopping
F:\JAVA - ABHI (MCA)>

Cp7 Create a package personpackage.This package contains a class Person with data members to represent firstName and lastName of a person and appropriate methods to read and display the same. Define appropriate class to test the above class outside the above package?

Solution

```
package personpackage;
import java.util.*;
class Person
{
   String firstName="Default",lastName="default";
   public void read(String fn,String ln)
   {
      firstName=fn;
      lastName=ln;
   }
   public void Display()
   {
      System.out.println("\n Name "+firstName+" "+lastName);
   }
}
public class Cp7 {
   public static void main(String[]args)
   {
      Person P1=new Person();
      Person P2=new Person();
      P1.read("Jack","Ryan");
      P1.Display();
      P2.read("Paul","Walker");
      P2.Display();
   }
}
```

Output

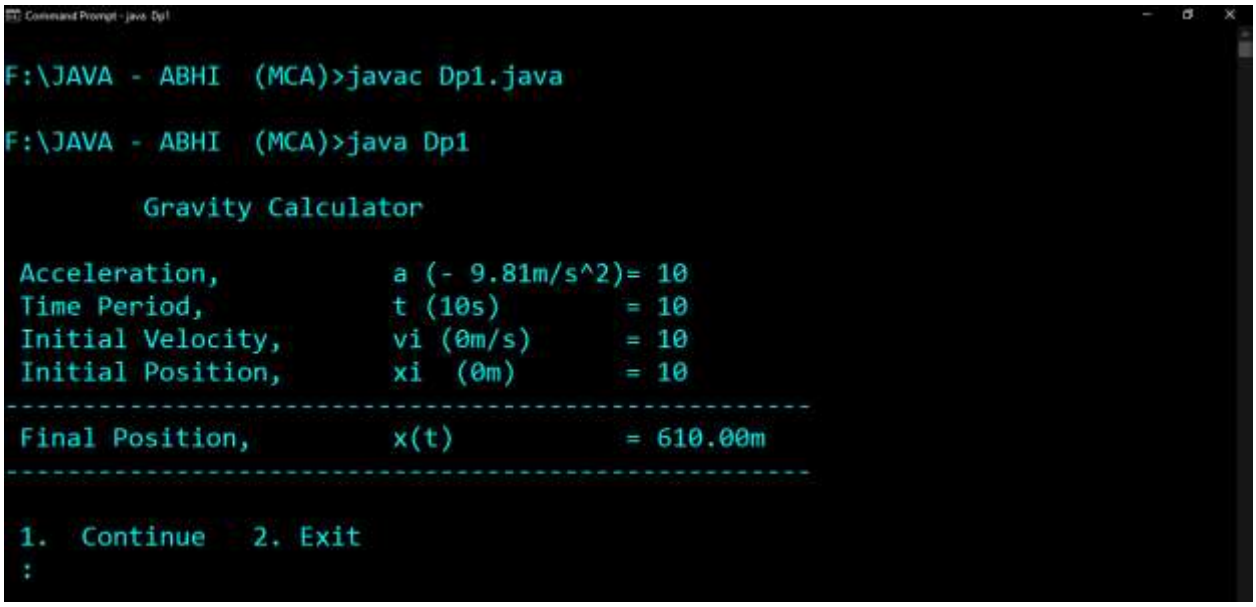

Command Prompt
F:\JAVA - ABHI (MCA)>javac Cp7.java
F:\JAVA - ABHI (MCA)>javac -d . Cp7.java
F:\JAVA - ABHI (MCA)>java personpackage.Cp7
Name Jack Ryan
Name Paul Walker
F:\JAVA - ABHI (MCA)>

Dp1 Extend your Gravity Calculator code (Assignment AA1) to handle exceptions through a try-catch finally block. Handle provision for a divide by zero scenarios caught by numberFormatException. Explicitly invoke this exception in execution and observe the response.

Solution

```
import java.util.*;
import java.text.*;
public class Dp1
{
   public static void main(String[] args)
   {
   try{
      do {
      Scanner sc = new Scanner(System.in);
      DecimalFormat f = new DecimalFormat("##.00");
      System.out.print("\n\t Gravity Calculator\n");
      System.out.print("\n Acceleration,\t\t a (- 9.81m/s^2)= ");
      double a = sc.nextDouble();
      System.out.print(" Time Period,\t\t t (10s)\t= ");
      double t = sc.nextDouble();
      System.out.print(" Initial Velocity,\t vi (0m/s)\t= ");
      double vi = sc.nextDouble();
      System.out.print(" Initial Position,\t xi  (0m)\t= ");
      double xi = sc.nextDouble();
      double xt = (0.5 * (a * t * t) + vi * t + xi);
      String ans = "\n Final Position,\t x(t) \t\t= " + f.format(xt) + "m\n";
      System.out.print("-".repeat(52)+ans);
      System.out.print("-".repeat(52)+"\n\n 1.  Continue\t2. Exit\n :");
      int contiCh = sc.nextInt();
      switch (contiCh) {
      case 1:
      continue;
      case 2:
      System.exit(0);
      default:
      System.out.println("Invalid Choice");
   }
} while (true);
}
catch(NumberFormatException e1){
```

```
System.out.print("\n Divide by Zero Exception!");
}
}
}
```

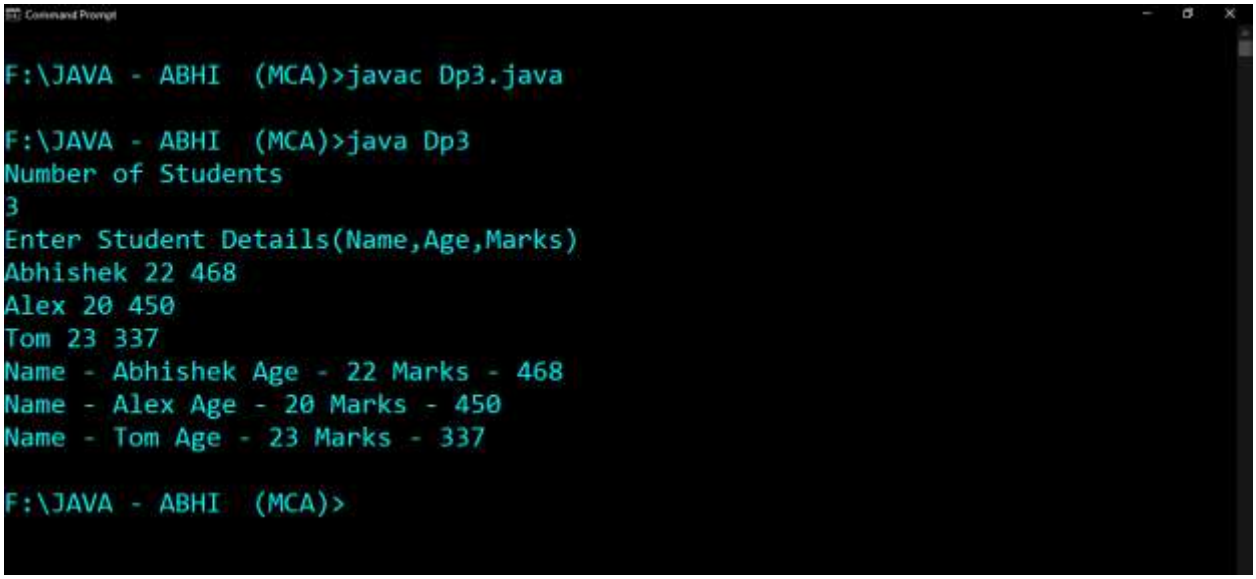


Dp3 Create a class StudentRegistrationCheck that checks eligibility of a student for registration. If the student age<12 and marks<200 then the student is not eligible for registration. Design appropriate solution using exception handling?

Solution

```
import java.util.*;
class StudentRegistrationCheck implements Comparable<StudentRegistrationCheck>
{
   private int marks;
   private String name;
   private int age;
   public StudentRegistrationCheck(String name,int age,int marks)
   {
      this.name=name;
      this.age=age;
      this.marks=marks;
   }
   public int compareTo(StudentRegistrationCheck o)
   {
      return this.name.compareTo(o.name);
   }
   public String toString()
   {
      return "Name - "+this.name + " Age - "+this.age+" Marks - "+this.marks;
   }
}
public class Dp3
{
   public static void main(String[]args)
   {
      TreeSet<StudentRegistrationCheck>stud=new TreeSet<>();
      Scanner sc=new Scanner(System.in);
      try
      {
         System.out.println("Number of Students ");
         int n=sc.nextInt();
         System.out.println("Enter Student Details(Name,Age,Marks)");
         for(int i=0;i<n;i++)
         {
            String name=sc.next();
            int age=sc.nextInt();
```

```
            int marks=sc.nextInt();
            if(age>=12 && marks>=200)
            {
               stud.add(new StudentRegistrationCheck(name, age, marks));
            }
          }
          for(StudentRegistrationCheck i:stud)
          {
            System.out.println(i);
          }
        }
        catch(Exception e1)
        {
          System.out.println("/n Exception !! Restart Program ");
        }
    }
}
```



Dp4 An ExceptionPOC class is requesting a number between 1 and 10. Run the program again and enter 5.5. Although this number is between 1 and 10, the program will abort. Examine the error message. You should see the word Exception, the method where the exception occurred (main), the class name of the exception (InputMismatchException), as well as the call stack listing the method calls.

Add a try/catch block to catch and handle the InputMismatchException exception. Identify the statements that cause the error as well as the portions of the program that depend upon these statements. Enclose these statements within the try block. Follow the try block with the catch block given below. Note, the InputMismatchException class is defined in java.util and must be imported. Also, when the Scanner throws an InputMismatchException, the input token will remain in the buffer so that it can be examined by the program. Complete code by implementing the same using:-

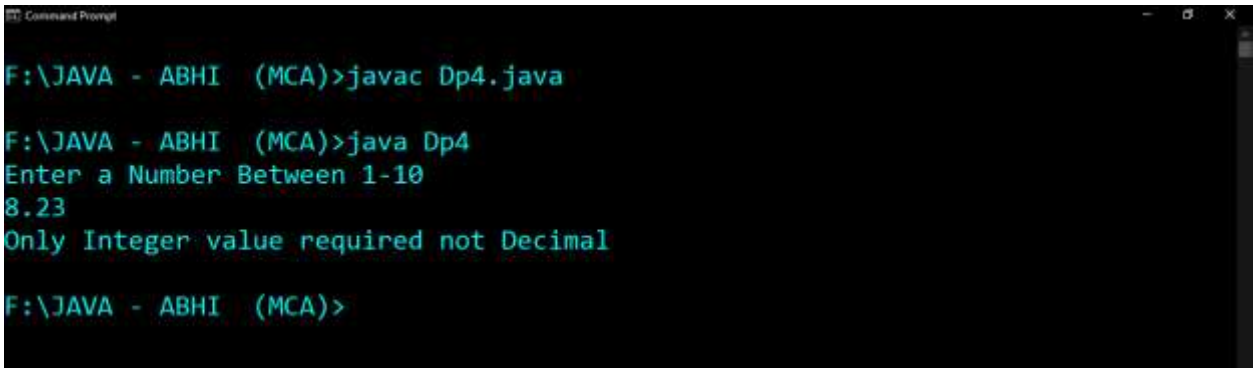
a) Throws method declaration

b) Throw keyword

Solution

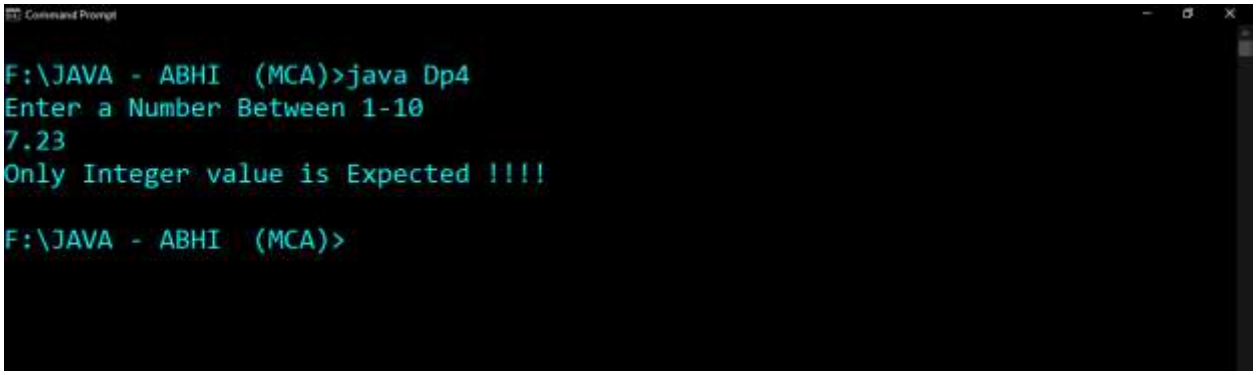
```
import java.util.*;
public class Dp4 {
   public static void main(String[]args)
   {
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter a Number Between 1-10");
      try{
         int i=sc.nextInt();
      }
      catch(InputMismatchException e)
      {
         System.out.println("Only Integer value required not Decimal");
      }
   }
```

```
}
```



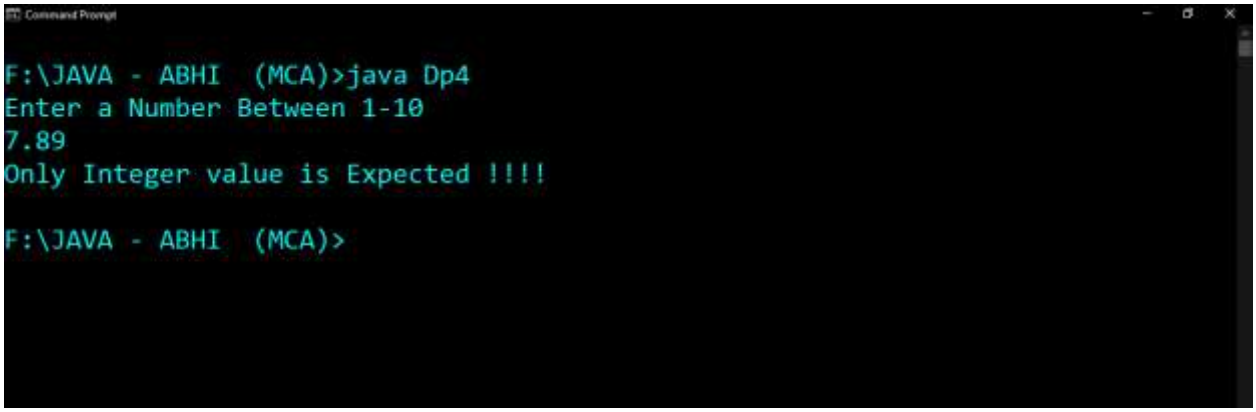
a)Throws method declaration

```
import java.util.*;
public class Dp4 {
public void fun() throws InputMismatchException
{
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter a Number Between 1-10");
         int i=sc.nextInt();
}
public static void main(String[]args)
{
   Dp4 obj=new Dp4();
   try{
      obj.fun();
   }
   catch(InputMismatchException e)
   {
      System.out.println("Only Integer value is Expected !!!! ");
   }
}
}
```



b)Throw Keyword

```
import java.lang.reflect.InaccessibleObjectException;
import java.util.*;
public class Dp4 {
public void fun() throws InputMismatchException
{
      Scanner sc=new Scanner(System.in);
      System.out.println("Enter a Number Between 1-10");
      try
      {
         int i=sc.nextInt();
      }
      catch(InputMismatchException e)
      {
         throw new InputMismatchException();
      }

}
public static void main(String[]args)
{
   Dp4 obj=new Dp4();
   try{
      obj.fun();
   }
   catch(InputMismatchException e)
   {
      System.out.println("Only Integer value is Expected !!!! ");
   }
}
}
```

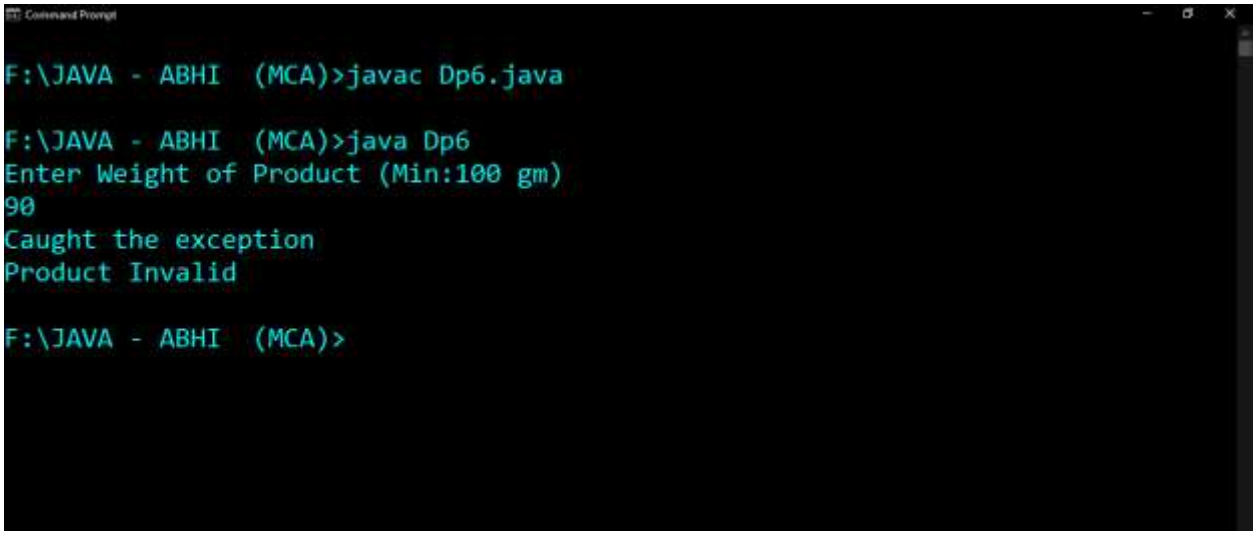


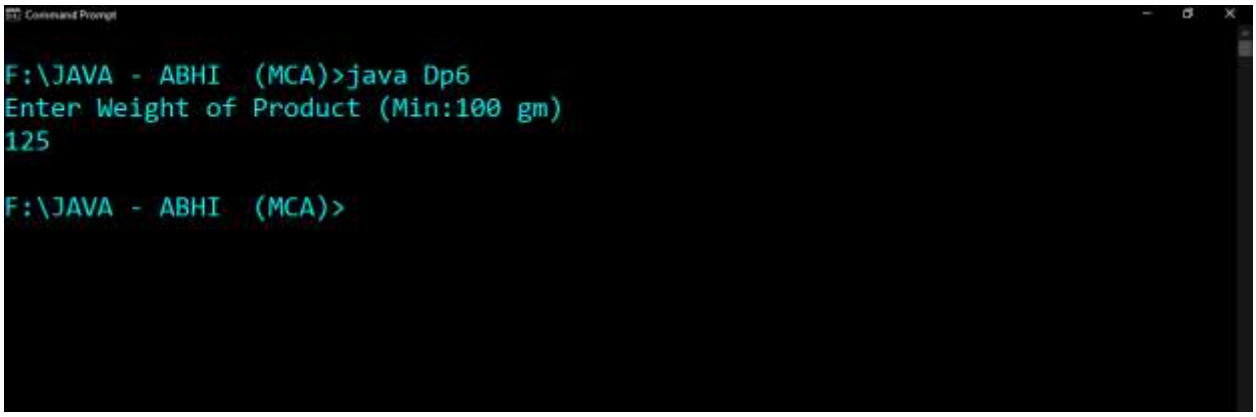
Dp6 Create an Exception class InvalidProductException that can be thrown if a user adds an invalid product

Solution

```
import java.util.*;
class InvalidProductException extends Exception
{
   public InvalidProductException(String s)
   {
      // Call constructor of parent Exception
      super(s);
   }
}

public class Dp6
{
  void productCheck(double weight) throws InvalidProductException{
          if(weight<100){
                  throw new InvalidProductException("Product Invalid");
          }
  }

   public static void main(String args[])
   {
          Dp6 obj = new Dp6();
      Scanner sc=new Scanner(System.in);
      try
      {
         System.out.println("Enter Weight of Product (Min:100 gm)");
         double w=sc.nextDouble();
         obj.productCheck(w);
      }
      catch (InvalidProductException ex)
      {
         System.out.println("Caught the exception");
         System.out.println(ex.getMessage());
      }
   }
}
```

Output


Command Prompt
F:\JAVA - ABHI (MCA)>javac Dp6.java
F:\JAVA - ABHI (MCA)>java Dp6
Enter Weight of Product (Min:100 gm)
90
Caught the exception
Product Invalid
F:\JAVA - ABHI (MCA)>


Command Prompt
F:\JAVA - ABHI (MCA)>java Dp6
Enter Weight of Product (Min:100 gm)
125
F:\JAVA - ABHI (MCA)>

Dp7 Compile and run BadThreads.java:

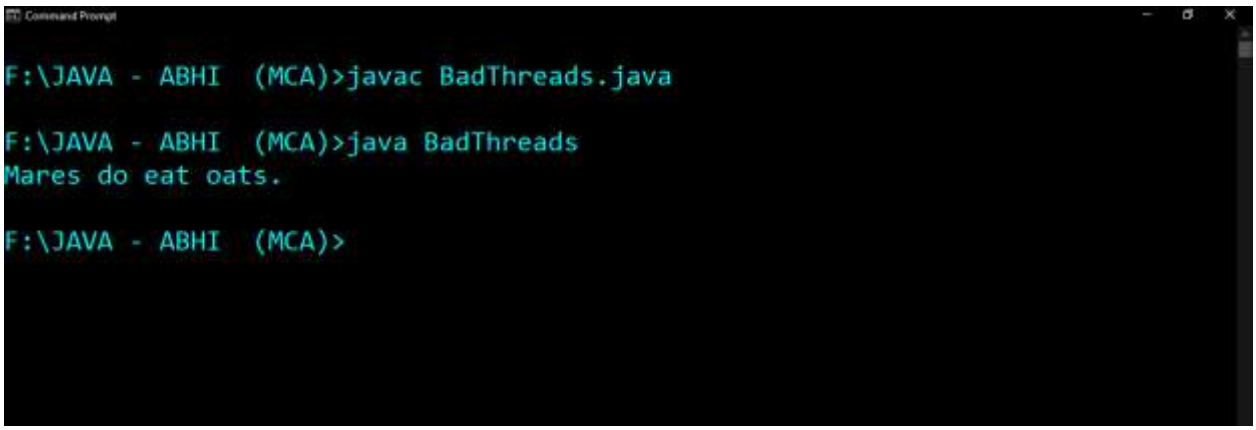
```
public class BadThreads {
static String message;
private static class CorrectorThread extends Thread {
public void run()
{ try
{ sleep(1000);}
catch (InterruptedException e) {}
        // Key statement 1:
        message = “Mares do eat oats.”;
    }}
public static void main(String args[]) throws InterruptedException {(new
CorrectorThread()).start();
message = “Mares do not eat oats.”;
Thread.sleep(2000);
// Key statement 2: System.out.println(message);
    }}
```

The application should print out “Mares do eat oats.” Is it guaranteed to always do this? If not, why not? Would it help to change the parameters of the two invocations of Sleep? How would you guarantee that all changes to the message will be visible in the main thread?

Solution

```
import java.util.*;
public class BadThreads
{
   static String message;
   private static class CorrectorThread extends Thread{
      public void run()
      {
         try
         {
            sleep(1000);
         }
         catch(InterruptedException e)
         {
```

```
        }
        // Key Statement 1:
        message="Mares do eat oats.";
      }
    }
    public static void main(String[]args) throws InterruptedException
    {
      (new CorrectorThread()).start();
      message="Mares do eat oats.";
      Thread.sleep(2000);
      System.out.println(message);
    }
}
```



Is it guaranteed to always do this? If not, why not?

Yes, but it is not guaranteed. This is because there is no happens-before relationship between 'Key Statement 1' and 'Key Statement 2'.

Would it help to change the parameters of the two invocations of Sleep?

It wouldn't help to change the parameters of the two invocations of Sleep since this does nothing to guarantee a happens-before relationship.

How would you guarantee that all changes to the message will be visible in the main thread?

There are two ways in which we can guarantee that all changes to the message will be visible in the main thread: 1. In the main thread, retaining a reference to the CorrectorThread instance. Then invoking join on that instance before referring to message. 2. Encapsulating message in an object with synchronized methods. Never referencing message though those methods.

Dp8 Implement the producer consumer problem using multithreading in java. In computing, the producer–consumer problem (also known as the bounded- buffer problem) is a classic example of a multi-process synchronization problem. The problem describes two processes, the producer and the consumer, which share a common, fixed-size buffer used as a queue.

a) The producer's job is to generate data, put it into the buffer, and start again.

b) At the same time, the consumer is consuming the data (i.e. removing it from the buffer), one piece at a time.

To make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer.

Solution

```
import java.util.*;
public class Dp8 {
	public static void main(String[] args)
		throws InterruptedException
	{
		final PC pc = new PC();

		// Create producer thread
		Thread t1 = new Thread(new Runnable() {
			@Override
			public void run()
			{
				try {
					pc.produce();
				}
				catch (InterruptedException e) {
					e.printStackTrace();
				}
			}
		});

		Thread t2 = new Thread(new Runnable() {
			@Override
			public void run()
			{
				try {
					pc.consume();
```

```
                        }
                        catch (InterruptedException e) {
                                e.printStackTrace();
                        }
                }
        });

        t1.start();
        t2.start();

        t1.join();
        t2.join();
}

public static class PC {

        LinkedList<Integer> list = new LinkedList<>();
        int capacity = 2;
        public void produce() throws InterruptedException
        {
                int value = 0;
                while (true) {
                        synchronized (this)
                        {
                                while (list.size() == capacity)
                                        wait();

                                System.out.println("Producer produced-"+ value);

                                list.add(value++);
                                notify();
                                Thread.sleep(1000);
                        }
                }
        }
        public void consume() throws InterruptedException
        {
                while (true) {
                        synchronized (this)
                        {
                                while (list.size() == 0)
                                wait();
                                int val = list.removeFirst();
                                System.out.println("Consumer consumed-"+ val);
                                notify();
                                Thread.sleep(1000);
                        }
```

```
                    }
                }
        }
}
```

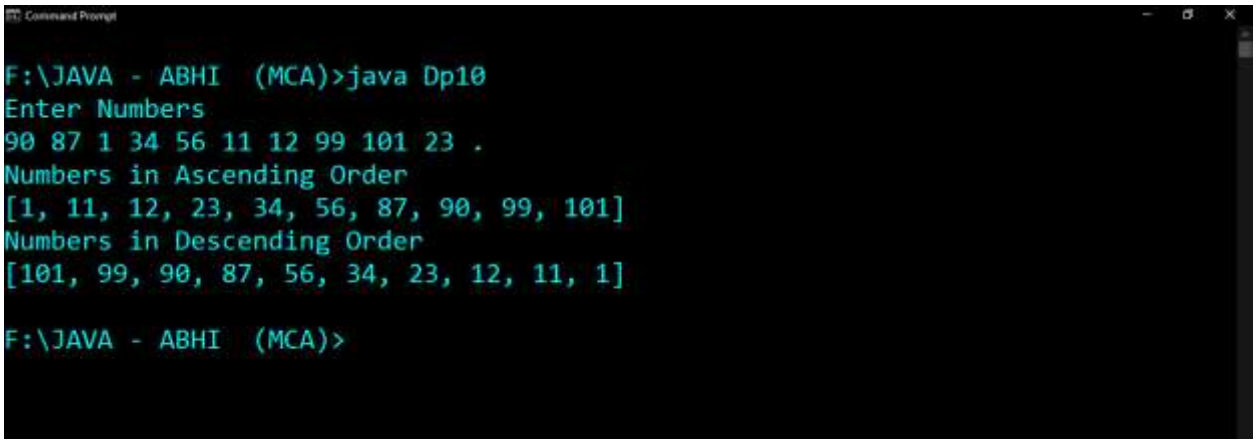
```
F:\JAVA - ABHI  (MCA)>javac Dp8.java

F:\JAVA - ABHI  (MCA)>java Dp8
Producer produced-0
Producer produced-1
Consumer consumed-0
Producer produced-2
Consumer consumed-1
Consumer consumed-2
Producer produced-3
Producer produced-4
```

Dp10 Create a list of numbers and then sort in ascending order as well as in descending order simultaneously.

Solution

```
import java.util.*;
public class Dp10 {
   public static void main(String[]args)
   {
      Scanner sc=new Scanner(System.in);
      List<Integer> intList=new ArrayList<Integer>();
      System.out.println("Enter Numbers ");
      while(sc.hasNextInt())
      {
         int i=sc.nextInt();
         intList.add(i);
      }
      Collections.sort(intList);
      System.out.println("Numbers in Ascending Order");
      System.out.println(intList);
      System.out.println("Numbers in Descending Order");
      Collections.sort(intList,Comparator.reverseOrder());
      System.out.println(intList);
   }
}
```

Output


Command Prompt
F:\JAVA - ABHI (MCA)>java Dp10
Enter Numbers
90 87 1 34 56 11 12 99 101 23 .
Numbers in Ascending Order
[1, 11, 12, 23, 34, 56, 87, 90, 99, 101]
Numbers in Descending Order
[101, 99, 90, 87, 56, 34, 23, 12, 11, 1]
F:\JAVA - ABHI (MCA)>

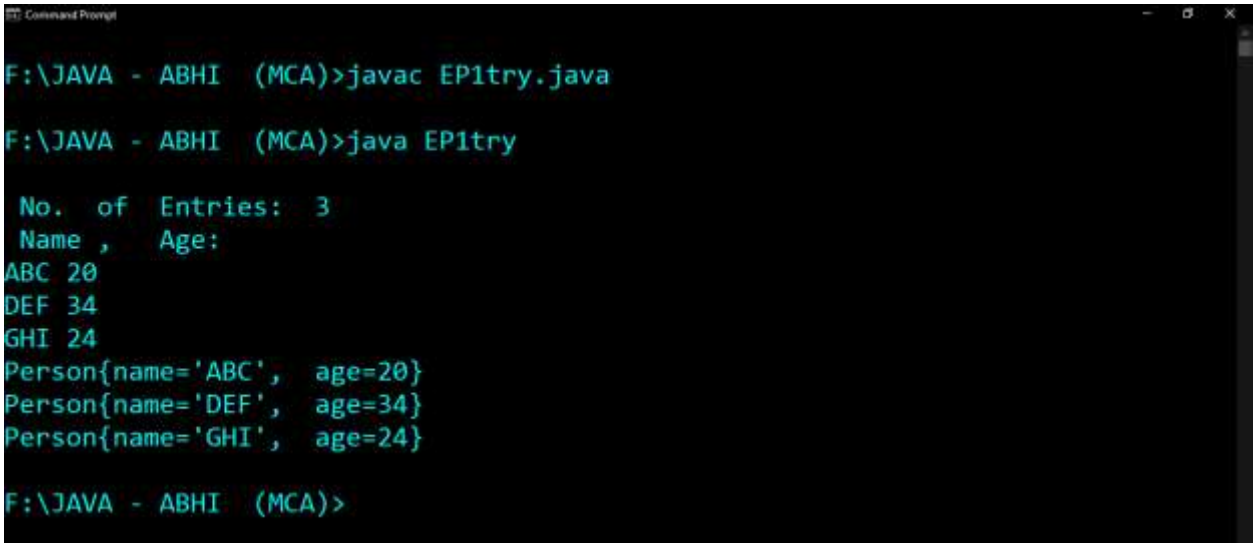
Ep1- Model Person with name and age.Manage instances of Person by ensuring that no two instances are duplicated?

Solution

```
import java.util.*;
class EP1Person implements Comparable<EP1Person>
{
String name;
int age;
EP1Person (String name, int age)
{ this.name = name;
this.age = age;
}
@Override
public int compareTo(EP1Person o)
{
   return this.name.compareTo(o.name);
}
@Override
public String toString()
{
  return "Person{" + "name='" + name + '\'' + ", age=" + age + '}';
}
}
public class EP1try
{

public static void main(String[] args)
{
   TreeSet<EP1Person> peeps = new TreeSet<>();
   Scanner sc = new Scanner(System.in);
   System.out.print("\n No. of Entries: ");
   int n = sc.nextInt();
   System.out.println(" Name ,  Age: ");
   for (int i = 0; i < n; i++) {
   String name = sc.next();
   int age = sc.nextInt();
   EP1Person p1 = new EP1Person(name, age);
   peeps.add(p1);
}
for (EP1Person i : peeps)
```

```
{
   System.out.println(i);
}
}
}
```

OUTPUT



Ep2- Create a subclass of Person (in EP1 above), called ComparablePerson which implements Comparable<Person> interface, and try out the Collections.sort() and Collections.binarySearch() methods on the same.

Solution:

```
import java.util.*;
class EP2Person implements Comparable<EP2Person>
{
 String name;
 int age;
 EP2Person (String name, int age)
 {
 this.name = name;
 this.age = age;
 }
 @Override
 public int compareTo(EP2Person o)
 {
 return this.name.compareTo(o.name);
 }
 @Override
 public String toString()
 {
 return " Person{" + "name='" + name + '\'' + ", age="
+ age + '}';
}
}
public class EP2
{
 static List<EP2Person> peeps = new
ArrayList<EP2Person>();
 static Scanner sc = new Scanner(System.in);
 static void prt()
 {
 for (EP2Person i : peeps)
 {
 System.out.println(i);
 }
 }
 static EP2Person inp()
 {
```

```
String name = sc.next();
int age = sc.nextInt();
EP2Person p1 = new EP2Person(name, age);
return p1;
}
public static void main(String[] args)
{
System.out.print("\n No. of Entries: ");
int n = sc.nextInt();
System.out.println(" Name, Age: ");
for (int i = 0; i < n; i++)
{
peeps.add(inp());
}
prt();
System.out.print("\n Collections.sort(): ");s
Collections.sort(peeps);
prt();
System.out.print("\n Collections.binarySearch():\n Enter Name, Age: ");
System.out.print(" Position:"+Collections.binarySearch(peeps, inp()));
}
}
```

OUTPUT


Command Prompt
Name, Age:
Alex 24
Tom 23
Harry 26
Jack 25
Person{name='Alex', age=24}
Person{name='Tom', age=23}
Person{name='Harry', age=26}
Person{name='Jack', age=25}
Collections.sort(): Person{name='Alex', age=24}
Person{name='Harry', age=26}
Person{name='Jack', age=25}
Person{name='Tom', age=23}
Collections.binarySearch():
Enter Name, Age: Jack 25
Position:2
F:\JAVA - ABHI (MCA)>

Ep3- Model AddressBookEntry that prints name,address and phone of a person. Allow comparison of AddressBookEntry to compare name in a case sensitive manner?

Solution

// AddressBookEntry.java

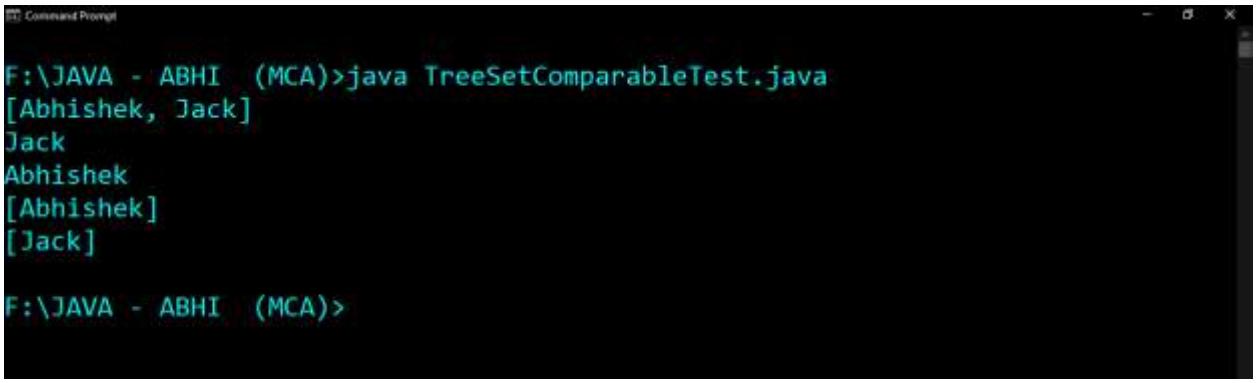
```
public class AddressBookEntry implements Comparable<AddressBookEntry>
 {
 private String name, address, phone;
 public AddressBookEntry(String name)
 {
 this.name = name;
 }
 @Override
 public String toString() {
 return name;
 }
 @Override
 public int compareTo(AddressBookEntry other) {
 return this.name.compareToIgnoreCase(other.name);
 }
 @Override
 public boolean equals(Object o) {
 return o != null && o instanceof AddressBookEntry &&
this.name.equalsIgnoreCase(((AddressBookEntry)o).name);
 }
 public int hashCode() {
 return name.toLowerCase().hashCode();
 }
}
```

// TreeSetComparableTest.java

```
import java.util.TreeSet;
public class TreeSetComparableTest {
 public static void main(String[] args) {
 AddressBookEntry addr1 = new AddressBookEntry("Abhishek");
 AddressBookEntry addr2 = new AddressBookEntry("Jack");
 TreeSet<AddressBookEntry> set = new TreeSet<>();
 set.add(addr1);
 set.add(addr2);
 System.out.println(set);
 System.out.println(set.floor(addr2));
```

```
 System.out.println(set.lower(addr2));
 System.out.println(set.headSet(addr2));
 System.out.println(set.tailSet(addr2));
 }
}
```

OUTPUT



Ep4- Counts the frequency of each of the words in a file given in the command-line,and save in a map of {word, freq}.

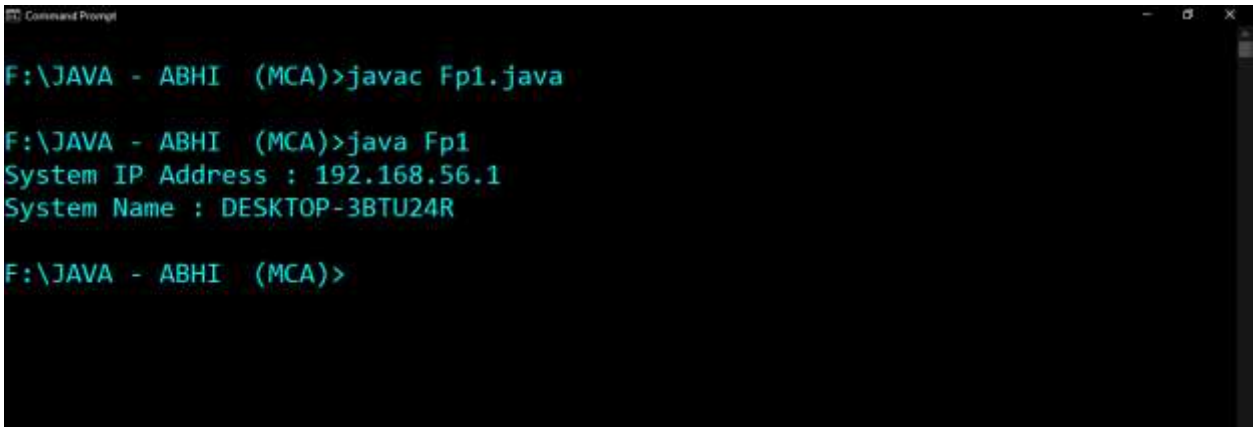
Solution

Fp1- Create a solution to know the address and name of your local machine?

Solution

```
import java.util.*;
import java.net.InetAddress;
public class Fp1{
 public static void main(String args[]) throws Exception{
 InetAddress localhost = InetAddress.getLocalHost();
 System.out.println("System IP Address : " +
(localhost.getHostAddress()).trim());
 try {
 String SystemName =
InetAddress.getLocalHost().getHostName();
 System.out.println("System Name : "+ SystemName);
 }
 catch (Exception E) {
 System.err.println(E.getMessage());
 }
 }
}
```

OUTPUT



Fp2- Create a solution to understand the different components of a URL?

Solution

```
import java.net.*;
import java.io.*;
import java.util.*;
import java.net.InetAddress;
public class Fp2 {
 // Main driver method
 public static void main(String[] args) throws Exception{
 URL url = new URL("https://www.richdad.com/about/rich-dad");
 System.out.println("URL is:" + url.toString());
 System.out.println("protocol is: "+
url.getProtocol());
 System.out.println("file name is: "+ url.getFile());
 System.out.println("host is: " + url.getHost());
 System.out.println("path is: " + url.getPath());
 System.out.println("port is: " + url.getPort());
 System.out.println("default port is: "+ url.getDefaultPort());
 }
}
```

OUTPUT

Fp3- Create a connection-less client/server application using UDP protocol that sends system date and time in the format requested by the client.

a) Client: Reads a string representing the required format from the end-user
b) Server:returns the system date and time in the requested format or a default format if received format is not understandable

Solution

// FP5.java

```
import java.net.*;
import java.io.*;
class Employee{
 private String EmpName;
 private int EmpId;
 private String EmpDept;
 public Employee(String EmpName, int EmpId, String EmpDept){
 this.EmpName = EmpName;
 this.EmpId = EmpId;
 this.EmpDept = EmpDept;
 }
 public String getEmpName(){
 return this.EmpName;
 }
 public int getEmpId(){
 return this.EmpId;
 }
 public String getEmpDept(){
 return this.EmpDept;
 }}
 class TCPIPServer {
 private Socket socket = null;
 private ServerSocket server = null;
 private DataInputStream in = null;
 private DataOutputStream out = null;
 public TCPIPServer(int port) {
 String id = ""; int i = 0;
 try {
   Employee[] empObj = new Employee[3];
   empObj[0] = new Employee("Abhishek ", 2," Programming");
   empObj[1] = new Employee("Hana", 4,"Robotics");
   empObj[2] = new Employee("Max", 3, "Dogs");
```

```
   server = new ServerSocket(port);
   System.out.println("Server started"); socket =
  server.accept();
   System.out.println("Client is now connected");
   in = new DataInputStream(new BufferedInputStream(socket.getInputStream()));
   out = new  DataOutputStream(socket.getOutputStream());
   id = in.readUTF();
    i = Integer.parseInt(id);
   for (int j = 0; j < empObj.length; j++)
   {
   if (empObj[j].getEmpId() == i)
   {
   int ids = empObj[j].getEmpId();
   String empId = Integer.toString(ids);
   String empName = empObj[j].getEmpName();
   String empDept = empObj[j].getEmpDept();
   String empInfo = "Emp Id = " + empId +  "\n" + "Emp Name = " + empName + "\n" + "Emp
Department = " + empDept;
   out.writeUTF(empInfo);
   break;
   }
   }
   in.close();
   out.close();
   socket.close();
   }
   catch (Exception e) {
   System.out.println("Exception found in TCPIPServer constructor" +e.getMessage());
   }
   }
   public static void main(String[] args) {
   TCPIPServer server = new TCPIPServer(5000);
   }
   }
```

// FP5C.java

```
import java.net.*;
import java.io.*;
class TCPIPClient{
 private Socket socket = null;
 private BufferedReader input = null; private
 DataInputStream in = null; private
 DataOutputStream out = null;
 public TCPIPClient(String address, int port){
 try{
 socket = new Socket(address,port);
 System.out.println("Conncted with server");
 input = new BufferedReader(new
```

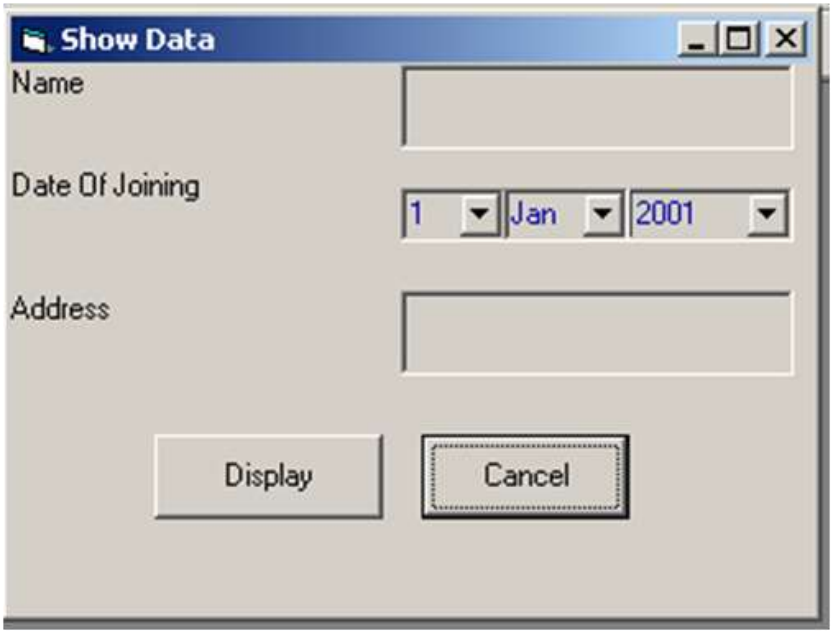
```
InputStreamReader(System.in)); out = new
 DataOutputStream(socket.getOutputStream());
 in = new
DataInputStream(socket.getInputStream());
 }
 catch(Exception e){
 System.out.println("Exception in TCP IPClient found "+e.getMessage());
 }
 String id = "";
 try{
 System.out.print("Enter employee id: ");
 id = input.readLine();
 out.writeUTF(id);
 }
 catch(Exception e){
 System.out.println("Exception in TCPIPClient user read value "+e.getMessage());
 }
 String empInfo = "";
 try {
 empInfo = in.readUTF();
System.out.println(empInfo);
 }
 catch(Exception e){
 System.out.println("Exception found in TCPIPClient empinfo "+e.getMessage());
 }
 try {
 in.close();
 out.close();
 input.close();
 socket.close();
 }
 catch(Exception e){
 System.out.println("Exception found in TCPIPClient closing "+e.getMessage());
 }
}
public static void main(String[] args){
TCPIPClient client = new TCPIPClient("localhost",5000);
}
}
```


Command Prompt - java TCPIPServer
F:\JAVA - ABHI (MCA)>javac Fp5.java
F:\JAVA - ABHI (MCA)>java TCPIPServer
Server started

Fp5- Create a calculator based client/server application where the client sends request to the server in the form of an arithmetic equation of the form "operand1 operator operator2". The server should respond back to answer the equation?

Solution

Gp3- Create the following layout using awt/swing



When user clicks the "Display" button the data entered by the user should be displayed in another frame window.

Solution

```
import javax.swing.*;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class GP3 {

 public static void main(String[] args) {

 UIDriver uiDriver = new UIDriver();

 uiDriver.createView();

 }
```

```
}

class UIDriver {

 SimpleDateFormat formatter = new SimpleDateFormat("dd-MMMyyyy");

 void createView() {

 JFrame frame = new JFrame("GP3 Abhishek Sharma 03911604421");

 JLabel label1, label2, label3;

 JTextField textField1, textField2;

 label1 = new JLabel("Name");

 textField1 = new JTextField("Enter your name.");

 label1.setBounds(10, 50, 60, 30);

 textField1.setBounds(100, 50, 200, 30);

 frame.add(label1);

 frame.add(textField1);

 label2 = new JLabel("Date of Joining");

 label2.setBounds(10, 100, 90, 30);

 frame.add(label2);

 String[] day = {"1", "2", "3", "4", "5", "6", "7","8", "9", "10","11", "12", "13", "14", "15", "16",
"17", "18", "19","20", "21", "22", "23", "24", "25", "26", "27", "28", "29","30", "31"};

 JComboBox<String> comboBoxDay = new JComboBox<>(day);

 comboBoxDay.setBounds(100, 100, 50, 30);

 frame.add(comboBoxDay);
```

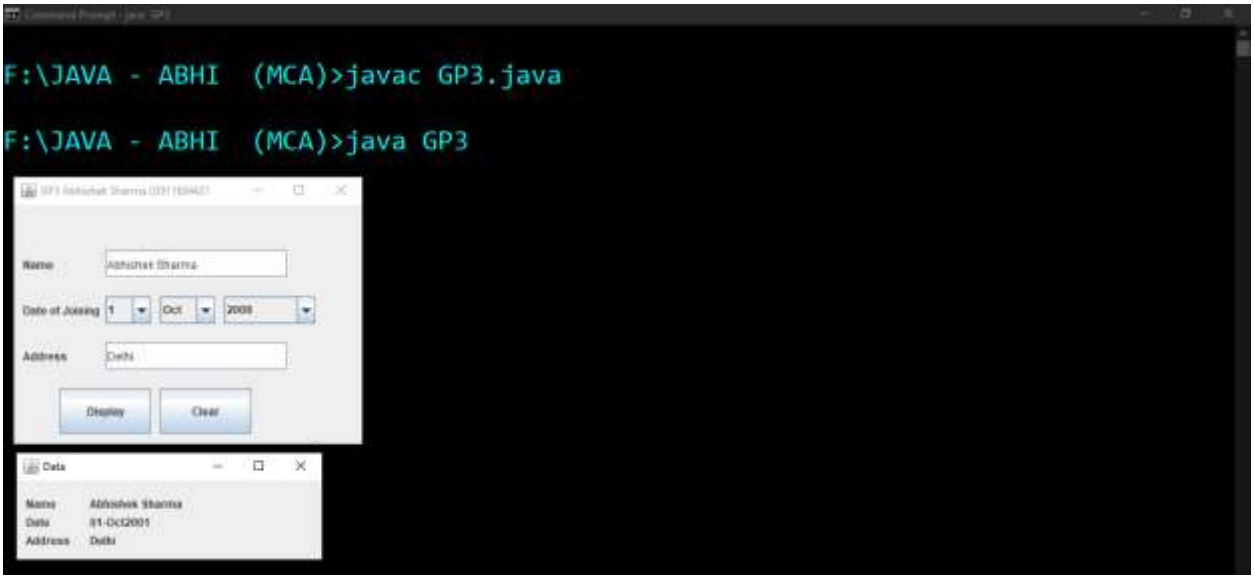
```
String[] month = {"Jan", "Feb", "Mar", "Apr", "May","Jun", "Jul", "Aug", "Sep", "Oct", "Nov",
"Dec"};

JComboBox<String> comboBoxMonth = new JComboBox<>(month);

comboBoxMonth.setBounds(160, 100, 60, 30);

frame.add(comboBoxMonth);

String[] year = new String[101];

for (int i = 0; i <= 100; i++)

{

year[i] = String.valueOf(2000 + i);

}

JComboBox<String> comboBoxYear = new JComboBox<>(year);

comboBoxYear.setBounds(230, 100, 100, 30);

frame.add(comboBoxYear);

label3 = new JLabel("Address");

textField2 = new JTextField("Enter your address.");

label3.setBounds(10, 150, 60, 30);

textField2.setBounds(100, 150, 200, 30);

frame.add(label3);

frame.add(textField2);

JButton display = new JButton("Display");

display.setBounds(50, 200, 100, 50);
```

```
display.addActionListener(e -> {
Date date = null;
String S = (comboBoxDay.getSelectedItem() + "-" + comboBoxMonth.getSelectedItem() + "-" +
comboBoxYear.getSelectedItem());
try {
date = formatter.parse(S);
}
catch (ParseException parseException) {
parseException.printStackTrace();
}
dataView(textField1.getText(),
textField2.getText(), date);
});
frame.add(display);
JButton cancel = new JButton("Clear");
cancel.setBounds(160, 200, 100, 50);
cancel.addActionListener(e -> {
frame.dispose();
createView();
});
frame.add(cancel);
```

```
frame.setSize(350, 300);

frame.setLayout(null);

frame.setVisible(true);

}

void dataView(String Name, String Address, Date date) {

JFrame frame = new JFrame("Data");

JLabel label1, label2, label3, label1Data, label2Data,label3Data;

label1 = new JLabel("Name");

label1.setBounds(10, 10, 60, 30);

frame.add(label1);

label1Data = new JLabel(Name);

label1Data.setBounds(80, 10, 200, 30);

frame.add(label1Data);

label2 = new JLabel("Date");

label2.setBounds(10, 30, 60, 30);

frame.add(label2);

label2Data = new JLabel(formatter.format(date));

label2Data.setBounds(80, 30, 200, 30);

frame.add(label2Data);

label3 = new JLabel("Address");
```

```
label3.setBounds(10, 50, 60, 30);

frame.add(label3);

label3Data = new JLabel(Address);

label3Data.setBounds(80, 50, 200, 30);

frame.add(label3Data);

frame.setSize(350, 125);

frame.setLayout(null);

frame.setVisible(true);

}

}
```

OUTPUT



Gp4 - Create a class MessageBox that extends Frame. The class should have a constructor that takes a String as a parameter to construct a dialog box that displays the message and OK & CANCEL buttons. The dialog box should get closed when the cancel buttons is clicked. Provide some mechanism in the MessageBox class that can be used by the calling program to check which button was pressed by the user. The class should have functions to:

a) Check which button was pressed by the user.

b) Retrieve the string entered by the user, if user pressed OK, null if user pressed CANCEL.

Test this class to get a message from a user and display it on a Frame window.

Solution

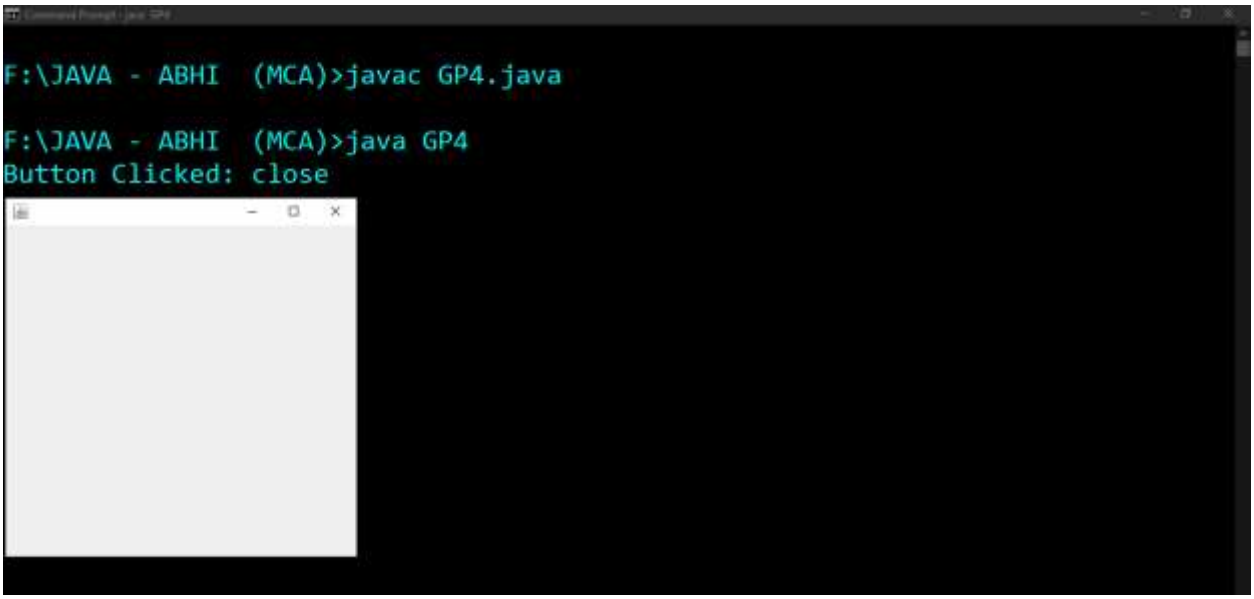
```
import java.awt.BorderLayout;

import java.awt.FlowLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.*;

import javax.swing.border.EmptyBorder;

public class GP4 {

 public static void main(String[] args) {



 MessageBox ob = new MessageBox("Assignment for OK and Cancel");

 ob.createView();
```

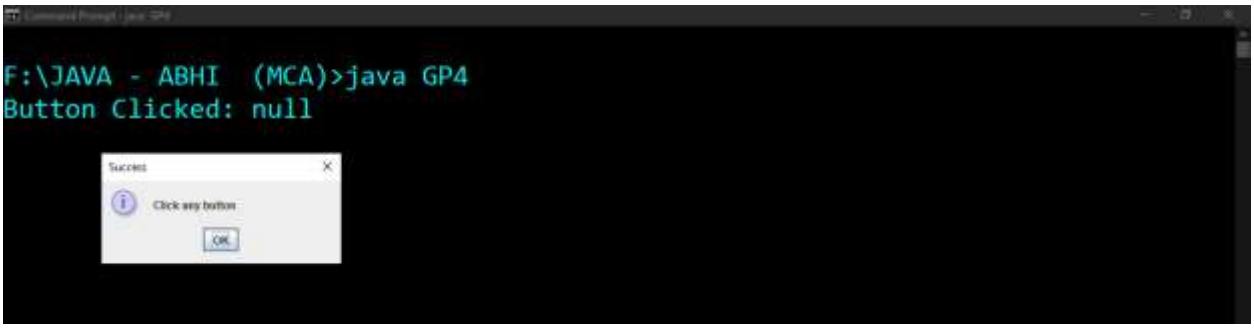
```
 System.out.println("Button Clicked: " + ob.getButtonClicked());
 }
}
class MessageBox extends JDialog implements ActionListener
{
 String message, buttonClicked;
 JButton accept, close;
 MessageBox(String message) {
 this.message = message;
 }
 void createView() {
   JFrame frame = new JFrame();
   frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
   frame.setSize(400, 400);
   JDialog dialog = new JDialog(frame, "User Prompt", true);
   JPanel mainGui = new JPanel(new BorderLayout());
   mainGui.setBorder(new EmptyBorder(20, 20, 20, 20));
   mainGui.add(new JLabel(message), BorderLayout.CENTER);
   JPanel buttonPanel = new JPanel(new FlowLayout());
   mainGui.add(buttonPanel, BorderLayout.SOUTH);
```

```
accept = new JButton("Accept");

accept.addActionListener(e -> {

dialog.setVisible(false);

JOptionPane.showMessageDialog(

null,

"Click any button",

"Success",

JOptionPane.INFORMATION_MESSAGE);

buttonClicked = "accept";

});

close = new JButton("Close");

close.addActionListener(e -> {

dialog.setVisible(false);

buttonClicked = "close";

});

buttonPanel.add(close);

buttonPanel.add(accept);

frame.setVisible(true);

dialog.setContentPane(mainGui);

dialog.pack();
```

```
dialog.setVisible(true);
}
@Override
public void actionPerformed(ActionEvent actionEvent)
{
}
public String getButtonClicked() {
return buttonClicked;
}
}
```

OUTPUT


F:\JAVA - ABHI (MCA)>javac GP4.java
F:\JAVA - ABHI (MCA)>java GP4
Button Clicked: close


F:\JAVA - ABHI (MCA)>java GP4
Button Clicked: null
Success
Click any button
OK

Hp1-A String tokenizer application to store the input string contents in a file. Read the file and count vowels, consonants and spaces in each line. Create another file to write the vowel and consonant count besides each line. For eg:- Hi this is java(vowels-5, consonents-7, spaces- 3). I like studying it(vowels-6, consonants-9, spaces-4). Perform this operation using:

a) BufferedReader and BufferedWriter

b) FileReader and FileWriter

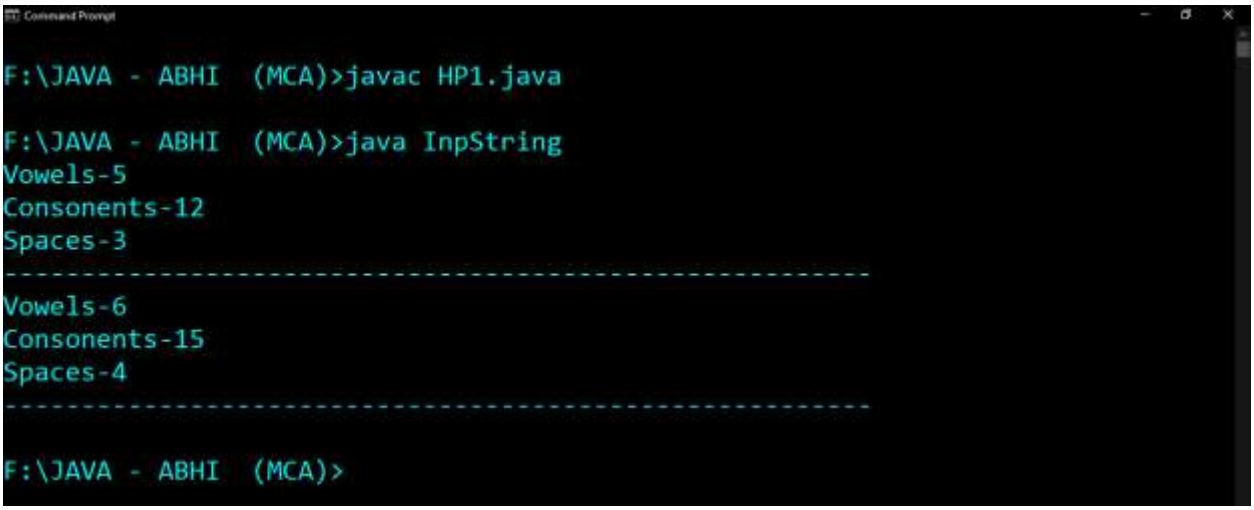
Solution

```
import java.util.*;

import java.io.*;

class InpString

{

 public void readVowelsConsonentsAndSpaces(String s){

 BufferedWriter bw = null;

 BufferedReader br = null;

 String s1=null;

 char ch=' ';

 int con=0;

 int vowels=0, letters=0, spaces=0;

 try

 {
```

```
bw = new BufferedWriter(new FileWriter(new File("test.txt")));
for(int i=0;i<s.length();i++){
con=(int)s.charAt(i);
bw.write(con);
}
bw.close();
}
catch(Exception e)
{
System.out.println("Exception found = "+e);
}
try
{
br = new BufferedReader(new
FileReader("test.txt"));
while((s1=br.readLine())!=null)
{
for(int i=0;i<s1.length();i++)
{
ch=s1.charAt(i);
```

```
if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'|| ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U')
{
   vowels++;
   letters++;
   }
   else if(ch==' ')
   {
   spaces++;
   }
   else
   {
   letters++;
   }
   }
   System.out.println("Vowels-"+vowels);
   System.out.println("Consonents-"+(letters));
   System.out.println("Spaces-"+spaces);
   System.out.println( "-----------------------------------------------------");
   vowels=0;
```

```
letters=0;

spaces=0;

}

br.close();

}

catch(Exception e){

System.out.println("Exception found = "+e);

}

}

public static void main(String[] args){

InpString sc = new InpString();

String s="Hi this is java\n I like studying it";

sc.readVowelsConsonentsAndSpaces(s);

}

}
```


Command Prompt
F:\JAVA - ABHI (MCA)>javac HP1.java
F:\JAVA - ABHI (MCA)>java InpString
Vowels-5
Consonents-12
Spaces-3
Vowels-6
Consonents-15
Spaces-4
F:\JAVA - ABHI (MCA)>

Hp-2 A File Parser a file and store the following text in it-

“Dwelling and speedily ignorant any steepest. Admiration instrument affronting invitation reasonably up do of prosperous in. Shy saw declared age debating ecstatic man. Call in so want pure rank am dear were. Remarkably to continuing in surrounded diminution on. In unfeeling existence objection immediate repulsive on he in. Imprudence comparison uncommonly me he difficulty diminution resolution. Likewise proposal differed scarcely dwelling as on raillery. September few dependent extremity own continued and ten prevailed attending. Early to weeks we could.

Unpleasant astonished an diminution up partiality. Noisy an their of meant. Death means up civil do an offer wound of. Called square an in afraid direct. Resolution diminution conviction so mr at unpleasing simplicity no. No it as breakfast up conveying earnestly immediate principle. Him son disposed produced humoured overcame she bachelor improved. Studied however out wishing but inhabit fortune windows.”

Accept a SearchToken from the user. Open the file and read it using RandomFileAccess and search and display total occurrences of the search string in given text.

Solution

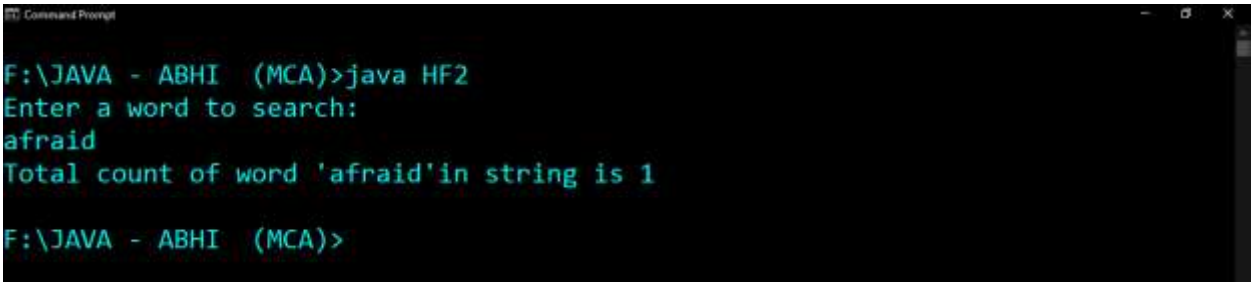
```
import java.util.Scanner;

import java.io.*;

public class HF2 {

 public void searchTok(String para){

 int count=0;
```

```
String st=null;
try{
FileWriter fw = new FileWriter("HF2.txt");
fw.write(para);
fw.close();
}
catch(Exception e){
System.out.println("Exception found = "+e);
}
Scanner sc = new Scanner(System.in);
System.out.println("Enter a word to search:");
String word = sc.nextLine();
try{
RandomAccessFile access = new RandomAccessFile(new File("HF2.txt"),"rw");
while((st=access.readLine())!=null)
{
String st1[] = st.split(" ");
for(String a:st1){
if(a.equals(word)){
count++;
```

```
}

}

}

System.out.println("Total count of word '"+word+"'in string is "+count);

access.close();

}

catch(Exception e){

System.out.println("Execption found = "+e);

}

}

public static void main(String[] args){

String para = "Dwelling and speedily ignorant any steepest.Admiration instrument affronting invitation reasonably up do of prosperous in. Shy saw declared age debating ecstatic man.Call in so want pure rank am dear were. Remarkably to continuing in surrounded diminution on. In unfeeling existence objection immediate repulsive on he in. Imprudence comparison uncommonly me he difficulty diminution resolution. Likewise proposal differed scarcely dwelling as on raillery. September few dependent extremity own continued and ten prevailed attending. Early to weeks we could. Unpleasant astonished an diminution up partiality. Noisy an their of meant. Death means up civil do an offer wound of. Called square an in afraid direct. Resolution diminution conviction so mr at unpleasing simplicity no. No it as breakfast up conveying earnestly immediate principle. Him son disposed produced humoured overcame she bachelor improved. Studied however out wishing but inhabit fortune windows.";

HF2 search = new HF2();

search.searchTok(para);
```

```
 }

}
```

OUTPUT


Command Prompt
F:\JAVA - ABHI (MCA)>java HF2
Enter a word to search:
afraid
Total count of word 'afraid'in string is 1
F:\JAVA - ABHI (MCA)>

HP3- Define your Student Class to Serialize objects of student class into separate files and byte streams using Serializable interface. Make use of the SerialVersionUID field and declare few variables as transient. Deserialize the objects and store them into an array of objects on another JVM instance and perform sorting based on parameter of user's choice using Comparator.

Note: You will have to make separate comparator classes for different data member based sorting.

Solution

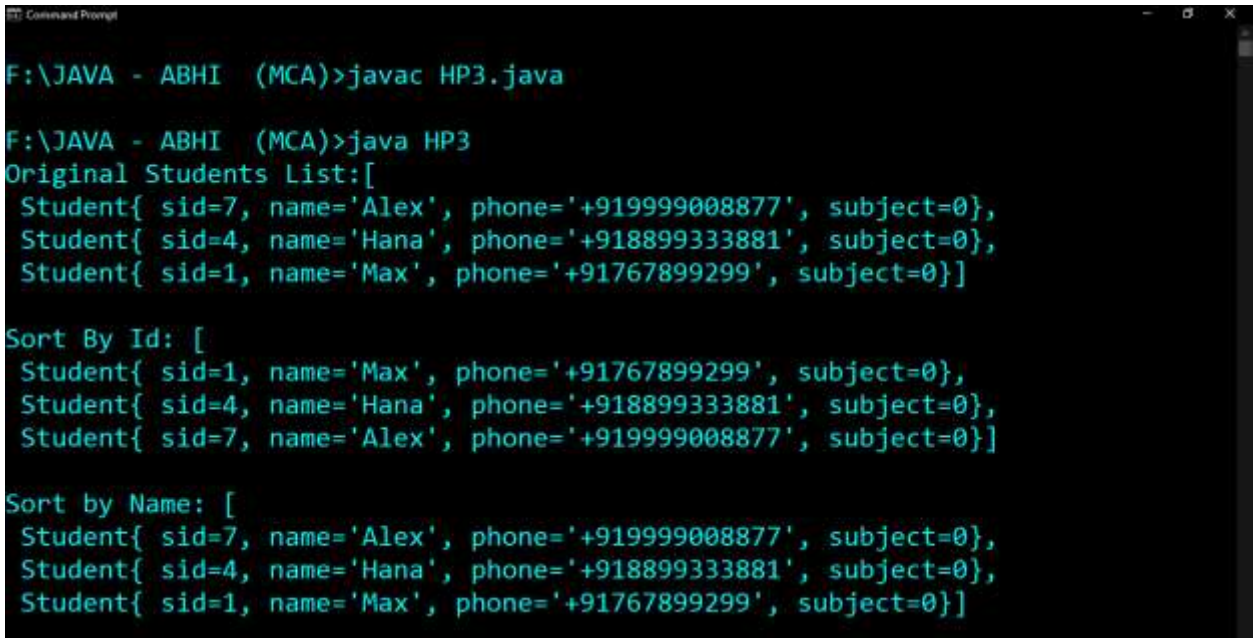
```
import java.io.*;

import java.util.*;

class Student implements Serializable{

 private static final long SerialVersion = 101;

 int sid;

 String name;

 String phone;

 transient int subject=5;

 Student(){}

 Student(int sid,String name, String phone){

 this.sid = sid;

 this.name = name;

 this.phone = phone;
```

```
 subject = 6;
 }
 @Override
 public String toString() {
 return "\n Student{" + " sid=" + sid + ", name='" + name + '\'' +  ", phone='" + phone + '\'' + ",
subject=" + subject + "}";
 }
}
class IdSort implements Comparator<Student>
{
 public int compare(Student o1, Student o2)
 {
 if(o1.sid < o2.sid)
 return -1;
 else
 return 1;
 }
}
class NameSort implements Comparator<Student>{
 public int compare(Student o1, Student o2){
 return (o1.name.compareTo(o2.name));
```

```
 }
}
class PhoneSort implements Comparator<Student>{
 public int compare(Student o1, Student o2){
 return (o1.phone.compareTo(o2.phone));
 }
}
public class HP3 {
 public static void main(String[] args) throws Exception {
 Student s1=new Student(7,"Alex", "+919999008877");
 Student s2=new Student(4,"Hana", "+918899333881");
 Student s3=new Student(1,"Max", "+91767899299");
 FileOutputStream fos = new FileOutputStream( "F:\\JAVA - ABHI  (MCA)\\HP3.txt");
 ObjectOutputStream oos=new ObjectOutputStream(fos);
 oos.writeObject(s1); oos.writeObject(s2);
oos.writeObject(s3);
 oos.close(); fos.close();
 List<Student> studentList=new ArrayList<>();
 Student s1Deserialized;
 Student s2Deserialized;
```

```
Student s3Deserialized;
 FileInputStream fis = new FileInputStream( "F:\\JAVA - ABHI  (MCA)\\HP3.txt");
 ObjectInputStream ois = new ObjectInputStream(fis);
 s1Deserialized=(Student)ois.readObject();
 s2Deserialized=(Student)ois.readObject();
 s3Deserialized=(Student)ois.readObject();
 studentList.add(s1Deserialized);
studentList.add(s2Deserialized);
 studentList.add(s3Deserialized);
 System.out.println("Original Students List:"+studentList);
 Collections.sort(studentList, new IdSort());
 System.out.println("\nSort By Id: "+ studentList);
 Collections.sort(studentList, new NameSort());
 System.out.println("\nSort by Name: "+ studentList);
 Collections.sort(studentList, new PhoneSort());
 System.out.println("\nSort by Phone: "+ studentList);
 }
}
```

OUTPUT


Command Prompt
F:\JAVA - ABHI (MCA)>javac HP3.java
F:\JAVA - ABHI (MCA)>java HP3
Original Students List:[
Student{ sid=7, name='Alex', phone='+919999008877', subject=0},
Student{ sid=4, name='Hana', phone='+918899333881', subject=0},
Student{ sid=1, name='Max', phone='+91767899299', subject=0}]
Sort By Id: [
Student{ sid=1, name='Max', phone='+91767899299', subject=0},
Student{ sid=4, name='Hana', phone='+918899333881', subject=0},
Student{ sid=7, name='Alex', phone='+919999008877', subject=0}]
Sort by Name: [
Student{ sid=7, name='Alex', phone='+919999008877', subject=0},
Student{ sid=4, name='Hana', phone='+918899333881', subject=0},
Student{ sid=1, name='Max', phone='+91767899299', subject=0}]


Command Prompt
Sort By Id: [
Student{ sid=1, name='Max', phone='+91767899299', subject=0},
Student{ sid=4, name='Hana', phone='+918899333881', subject=0},
Student{ sid=7, name='Alex', phone='+919999008877', subject=0}]
Sort by Name: [
Student{ sid=7, name='Alex', phone='+919999008877', subject=0},
Student{ sid=4, name='Hana', phone='+918899333881', subject=0},
Student{ sid=1, name='Max', phone='+91767899299', subject=0}]
Sort by Phone: [
Student{ sid=1, name='Max', phone='+91767899299', subject=0},
Student{ sid=4, name='Hana', phone='+918899333881', subject=0},
Student{ sid=7, name='Alex', phone='+919999008877', subject=0}]
F:\JAVA - ABHI (MCA)>