Java Programming [CSE201] Enrolment No.: 23DCS056

**CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY**

**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH**

Department of Computer Science & Engineering

**Subject Name: Java Programming**

**Semester: 3**

**Subject Code: 203**

**Academic year: 2024-2025**

Part - 1

|  |  |
| --- | --- |
| **No.** | **Aim of the Practical** |
| 1 | Demonstration of installation steps of Java,Introduction to Object Oriented Concepts, comparison of Java with other object-oriented programming languages. Introduction to JDK, JRE, JVM, Javadoc, command line argument. Introduction to Eclipse or NetBeans IDE,or BlueJ and Console Programming.  **Answer:**  **Java** is one of the most popular and widely used programming language and platform. A platform is an environment that helps to develop and run programs written in any programming language.Java is fast, reliable and secure. From desktop to web applications, scientific supercomputers to gaming consoles, cell phones to the Internet, Java is used in every nook and corner.  **Java Development Kit (JDK**) is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), and other tools needed in Java development.  Now we need an environment to make a run of our program. Henceforth, **JRE** stands for **“Java Runtime Environment”** and may also be written as **“Java RTE.”** The Java Runtime Environment provides the minimum requirements for executing a Java application; it consists of the *Java Virtual Machine (JVM), core classes*, and *supporting files*.  Now let us discuss **JVM**, which stands out for java virtual machines. It is as follows:   * A **specification** where the working of Java Virtual Machine is specified. But implementation provider is independent to choose the algorithm. Its implementation has been provided by Sun and other companies. * An **implementation** is a computer program that meets the requirements of the JVM specification. * **Runtime Instance** Whenever you write a java command on the command prompt to run the java class, an instance of JVM is created. |
| 2. | Imagine you are developing a simple banking application where you need to display the current balance of a user account. For simplicity, let's say the current balance is $20. Write a java program to store this balance in a variable and then display it to the user.  **PROGRAM CODE:**  Public class practical2{  Public static void main(String [] args){  Int balance;  System.out.println(“your balance is”+balance);  }  }  **OUTPUT:**    **CONCLUSION:**  This practical teaches us how to store and display data in the form of variables. Variables have the ability to store any type of data be it integer, float, long, character or even string. To store data we must first define the datatype first, then define a name for the variable. Then we can use the equal sign to store the value. |
| 3. | Write a program to take the user for a distance (in meters) and the time taken (as three numbers: hours, minutes, seconds), and display the speed, in meters per second, kilometers per hour and miles per hour (hint:1 mile = 1609 meters).  **PROGRAM CODE:**  import java.util.\*;  public class Practical3 {  public static void main(String[] args) {  Scanner cin = new Scanner(System.in);  System.out.println("enter the distance in meters");  float meter, kilometer, miles;  meter = cin.nextFloat();  kilometer = meter / 1000;  miles = meter/1609;  System.out.println("enter the time in hours");  int hours;  hours = cin.nextInt();  System.out.println("enter the time in minutes");  int minutes = cin.nextInt();  System.out.println("enter the time in seconds");  int seconds = cin.nextInt();  float totalsec,totalmin,totalhour;  totalsec = hours \* 3600 + minutes \* 60 + seconds;  totalmin = totalsec / 60;  totalhour = totalmin / 60;    System.out.println("totalsec" + totalsec);  System.out.println("totalmin"+totalmin);  float mps, kmh, mph;  mps = meter / totalsec;  kmh = kilometer / totalhour;  mph = miles / totalhour;  System.out.println("the speed in meters/second is " + mps);  System.out.println("the speed in kilometers/hour is " + kmh);  System.out.println("the speed in miles/hour is " + mph);  cin.close();  }  }  **OUTPUT:**    **CONCLUSION:**  By this practical we learnt to take input from console and perform arithmetic calculations on variables to find the speed of a vehicle in different parameters. The datatypes used were float and integer. To convert seconds, minutes and hours into hours, we converted the seconds and minutes into hours and added those to the stored hours. |
| 4. | Imagine you are developing a budget tracking application. You need to calculate the total expenses for the month. Users will input their daily expenses, and the program should comput the sum of these expenses. Write a Java program to calculate the sum of elements in an array representing daily expenses.  **PROGRAM:**  import java.util.\*;  public class Practical4 {  public static void main(String[] args) {  int[] expense;  expense = new int[30];  Scanner cin = new Scanner(System.in);  int choice;  System.out.println("how many expenses");  choice = cin.nextInt();  for (int i = 0; i < choice; i++) {  System.out.println("enter the expense" + (i + 1));  expense[i] = cin.nextInt();  }    int sum=0;  for (int i = 0; i < choice; i++) {  sum += expense[i];  }  System.out.println("your total expense " + sum);  cin.close();  }  }  **OUTPUT:**    **CONCLUSION:**  By this practical, we learnt to use for loops and arrays in java. To define an array we have to declare it first and then we have to allocate some memory for it using the ‘new’ operator then we can start assigning it some values. To initiate the for loop we need a counter variable and a condition. |

|  |  |
| --- | --- |
| 4. | **Supplementary Experiment**:  You are creating a library management system. The library has two separate lists of books for fiction and non-fiction. The system should merge these lists into a single list for inventory purposes. Write a Java program to merge two arrays.  **PROGRAM:**  import java.util.\*;  public class Practical4sup {  public static void main(String[] args) {  String[] fiction, non\_fiction, merged;  Scanner sc = new Scanner(System.in);  int i,j;  fiction = new String[5];  non\_fiction = new String[5];  System.out.println("enter the elements in fictional section");  for (i = 0; i < 5; i++) {  System.out.println("enter Element" + (i + 1));  fiction[i] = sc.nextLine();  }  System.out.println("enter the elements in non fictional section");  for (i = 0; i < 5; i++) {    System.out.println("enter Element" + (i + 1));  non\_fiction[i] = sc.nextLine();  }  merged = new String[10];    for (i = 0; i < 5; i++) {  merged[i] = fiction[i];  }  j=i;  int k=0;  for (j = i; j < 10; j++) {  merged[j] = non\_fiction[k];  k++;  }  System.out.println("arrays merged!");  System.out.print("resultant array: ");  for (i = 0; i < 10; i++) {  System.out.println(merged[i]);  }  sc.close();    }  }  **OUTPUT:**    **CONCLUSION:**  This program teaches us how to merge two arrays using indexing. With the help of for loops and indexing we can merge two arrays and even sort them. It also teaches us the importance of closing the scanner object we create for taking input. To do so we must use the .close() attribute. |
| 5 | An electric appliance shop assigns code 1 to motor,2 to fan,3 to tube and 4 for wires. All other items have code 5 or more. While selling the goods, a sales tax of 8% to motor,12% to fan,5% to tube light,7.5% to wires and 3% for all other items is charged. A list containing the product code and price in two different arrays. Write a java program using switch statement to prepare the bill.  **PROGRAM:**  import java.util.Scanner;  public class Practical5\_1 {  public static void main(String[] args) {  int[] price = { 200, 3000, 100, 50, 80 };  int[] code = { 1, 2, 3, 4, 5 };  String[] item = { "Motor", "Fan", "Tube", "Wire", "Others" };  Scanner sc = new Scanner(System.in);  System.out.println("Welcome to Bijli Appliances. What do you want to buy?");    int choice;  int[] qty = { 0, 0, 0, 0, 0 };    for (int i = 0; i < 5; i++) {  System.out.println(code[i] + "." + item[i] + "\t" + price[i]);  }  System.out.println("To exit the program press 6");  do {    System.out.print("Enter your choice: ");  choice = sc.nextInt();    switch (choice) {  case 1:  case 2:  case 3:  case 4:  case 5:  qty[choice - 1] += 1;  break;  case 6:  System.out.println("Exiting...");  break;  default:  System.out.println("Invalid choice, please select a valid option.");  }    } while (choice != 6);    System.out.println("\nThis is your Bill:");  System.out.println("\tBijli Appliances");    int totalGoods = 0;  int totalPrice = 0;    for (int i = 0; i < 5; i++) {  if (qty[i] > 0) {  System.out.println(item[i] + "\tQuantity: " + qty[i] + "\tPrice: " + (qty[i] \* price[i]));  totalGoods += qty[i];  totalPrice += qty[i] \* price[i];  }  }    System.out.println("\nTotal Qty of Goods: " + totalGoods);  System.out.println("Total Bill: " + totalPrice);  System.out.println("Thanks for visiting... come again!");  sc.close();  }  } |
|  | **OUTPUT:**    **CONCLUSION:**  This program teaches us how to use arrays and conditional statements to solve real life applications such as electrical store bill. In this we use switch statements and array indexing as well as a do while loop to prepare the bill |
| 6 | Create a Java program that prompts the user to enter the number of days (n) for which they want to generate their exercise routine. The program should then calculate and display the first n terms of the Fibonacci series, representing the exercise duration for each day.  **PROGRAM:**  import java.util.\*;  public class Practical6 {  public static void main(String[] args) {  Scanner cin = new Scanner(System.in);  System.out.println("Welcome to the exercise program!");  System.out.println("enter the number of days you want to exercise for:");  int n = cin.nextInt();  System.out.println("This is your exercise schedule");  for (int i = 1; i < n; i++) {  System.out.println("Day "+i+":You have to exercise for "+fib(i)+" hours");  }    cin.close();  }    public static int fib(int n) {  if (n == 1 || n == 0) {  return n;  }  else {  return fib(n - 1) + fib(n - 2);  }  }  }  **OUTPUT:**    **CONCLUSION:**  By this program we learnt how to implement the Fibonacci series using recursive functions and solve real life scenario of exercising schedule. |
| 6 | **Supplementary Experiment:**  Imagine you are developing a classroom management system. You need to keep track of the grades of students in a class. After collecting the grades, you want to display each student's grade along with a message indicating if they have passed or failed. Let's assume the passing grade is 50.  **PROGRAM:**  import java.util.\*;  public class Practical6sup {  /\*\*  \* @param args  \*/  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  int[] roll, grades;  roll = new int[10];  for(int i=1;i<=10;i++){  roll[i-1]=i;  }  grades = new int[10];  System.out.println("Welcome to the Classroom!");  System.out.println("What would you like to do");  int choice;  do{  System.out.println("1.Enter Student Grades");  System.out.println("2.Display Student Grades");  System.out.println("3.Exit the program");  System.out.println("enter your choice:");  choice = sc.nextInt();  if(choice==1){  for(int i=1;i<=10;i++){  System.out.println("Enter the Grade for roll no. "+i+":");  grades[i-1]=sc.nextInt();  }  System.out.println("Grades recorded and saved successfully!");  }  else{  for(int i=1;i<=10;i++){  System.out.println("Grades for Roll no. "+i+" are "+grades[i-1]);  if(grades[i-1]>=50){  System.out.println("The student has passed");  }  else{  System.out.println("The student has failed.");  }  }  }    } while (choice != 3);  System.out.println("Thanks for using this program!");  sc.close();  }  }  **OUTPUT:**    **CONCLUSION:**  By this Experiment we learned how to use arrays and conditional statements to develop a student management console. We stored some values in an array and used traversing to display them and checked each element for passing condition of a student. |
| 7 | Given a string and a non-negative int n, we'll say that the front of the string is the first 3 chars, or whatever is there if  the string is less than length 3. Return n copies of the front  front\_times('Chocolate', 2) → 'ChoCho'  front\_times('Chocolate', 3) → 'ChoChoCho'  front\_times('Abc', 3) → 'AbcAbcAbc'  **PROGRAM:**  import java.util.\*;  public class Practical7 {  public static void main(String[] args) {  String input;  int n;  Scanner sc = new Scanner(System.in);  input = sc.nextLine();  n=sc.nextInt();  front\_times(input, n);  sc.close();  }    public static void front\_times(String S, int n) {  for (int i = 0; i < n; i++) {  if (S.length() < 3) {  System.out.print(S);  }  else{    System.out.print(S.substring(0, 3));  }  }  }  }  **OUTPUT:**    **CONCLUSION:**  In this example, we learn how to manipulate strings in java and use the concept of substrings and repeated a substring multiple times using a userdefined function. |
| 8 | Given an array of ints, return the number of 9's in the  array. array\_count9([1, 2, 9]) → 1  array\_count9([1, 9, 9]) → 2  array\_count9([1, 9, 9, 3, 9]) → 3  **PROGRAM:**  public class Practical8 {      public static void main(String[] args) {          int[] arr1 = { 1, 2, 9 };          int[] arr2 = { 1, 9, 9 };          int[] arr3 = { 1, 9, 9, 3, 9 };            System.out.println(array\_count9(arr1));          System.out.println(array\_count9(arr2));          System.out.println(array\_count9(arr3));      }        public static int array\_count9(int[] n) {          int count=0;          int length= n.length;          for (int i = 0; i < length; i++) {              if (n[i] == 9) {                  count += 1;              }            }          return count;      }  }  **OUTPUT:**    **CONCLUSION:**  In this example we learnt how to manipulate arrays and perform operations on them such as finding how many times a single element gets repeated, in this case we had to search for ‘9’. |
| 8 | **Supplementary Experiment:**  Write a Java program to replace each substring of a given string that matches the given regular expression with the given replacement.  Sample string : "The quick brown fox jumps over the lazy dog."  **In the above string replace all the fox with cat.**  **PROGRAM:**  public class Practical8sup {          public static void main(String[] args) {              String sampleString = "The quick brown fox jumps over the lazy dog.";              String regex = "fox";              String replacement = "cat";                String resultString = sampleString.replaceAll(regex, replacement);                System.out.println("Original String: " + sampleString);              System.out.println("Modified String: " + resultString);          }  }    **OUTPUT:**    **CONCLUSION:**  in this example we learnt how to use String class methods to solve problems which look difficult without those methods. One of the methods is known as replaceall. |
| 9 | Given a string, return a string where for every char in the original, there are two chars.  double\_char('The') → 'TThhee'  double\_char('AAbb') → 'AAAAbbbb'  double\_char('Hi-There') → 'HHii--TThheerree'  **PROGRAM:**  public class Practical9 {              public static void main(String[] args) {              System.out.println(doubleChar("The"));              System.out.println(doubleChar("AAbb"));              System.out.println(doubleChar("Hi-There"));          }            public static String doubleChar(String str) {              StringBuilder result = new StringBuilder();              for (int i = 0; i < str.length(); i++) {                  char c = str.charAt(i);                  result.append(c).append(c);              }              return result.toString();          }    }  **OUTPUT:**    **CONCLUSION:**  This example teaches us how to manipulate strings in way that each character of the given string is repeated a number of times. For that we have to use the existing for loops and use some string methods.. |
| 10 | Perform following functionalities of the string:   * Find Length of the String * Lowercase of the String * Uppercase of the String * Reverse String * Sort the String   **PROGRAM:**  import java.util.\*;  public class Main {  public static void main(String[] args) {  String s1="CHARUSAT UNIVERSITY";  System.out.println(s1.length());    System.out.println(s1.toUpperCase());  System.out.println(s1.toLowerCase());    String s1rev;  s1rev=Reverse(s1);    System.out.println(s1rev);  String sorted;    sorted= sort(s1);  System.out.println(sorted);    System.out.println("23dcs056\_Vansh\_Malani");    }    public static String Reverse(String str){  StringBuilder sb = new StringBuilder(str);  return sb.reverse().toString();  }      public static String sort(String str){  char temp[]=str.toCharArray();    Arrays.sort(temp);    return new String(temp);    }  }  **OUTPUT:**    **CONCLUSION:**  This example introduced us to String Builder Class and its methods and functions and how to apply them and convert them to string. it also taught us how to convert a string object to a character array in order to reverse it. |
| 11 | Perform following Functionalities of the string:  “CHARUSAT UNIVERSITY”  ● Find length  ● Replace ‘H’ by ‘FIRST LETTER OF YOUR NAME’  ● Convert all character in lowercase  **PROGRAM:**  public class Practical11{  public static void main(String args[]){  String s1="CHARUSAT UNIVERSITY";  System.out.println(s1.length());  String replaceString=s1.replace('H','V');//replaces all occurrences of 'a' to 'e'  System.out.println(replaceString);  System.out.println(s1.toLowerCase());  System.out.println("23dcs056\_vansh\_malani");  }}  **OUTPUT:**    **CONCLUSION:**  This example teaches us how to use the replace method of string class to replace individual characters of a string and convert the string to lowercase. |
| 11 | Supplementary Experiment:  1. Write a Java program to count and print all duplicates in  the input string.  Sample Output:  The given string is: resource  The duplicate characters and counts are:  e appears 2 times  r appears 2 times  **PROGRAM:**  import java.util.\*;  public class Practical11sup{  public static void printDuplicates(String str)  {  int len = str.length();  // Sorting the string  char[] chars = str.toCharArray();  Arrays.sort(chars);  String sortedStr = new String(chars);  // Loop through the sorted string to find duplicates  for (int i = 0; i < len; i++) {  int count = 1;  // Counting the occurrences of each character  while (i < len - 1 && sortedStr.charAt(i)== sortedStr.charAt(i + 1)) {  count++;  i++;  }  // Printing the duplicate character and its  // count  if (count > 1) {  System.out.println(sortedStr.charAt(i)  + ", count = " + count);  }  }  }  public static void main(String[] args)  {  String str = "resources";  System.out.println(str);  System.out.println("the duplicates in this string are:");  printDuplicates(str);  }  }  **OUTPUT:**    **CONCLUSION:**  In this example we learnt how to loop for duplicate characters in a string object, how to sort every character and find the frequency of each character. |