|  |  |
| --- | --- |
| S. No. | **Note: First few programs you have to complete without Scanner class.** |
|  | Write a program that produces the following output:  Hello World!  It's been nice knowing you.  Goodbye world! |
|  | State the order of evaluation of the operations in each of the following Java statements and implement them to show the value of x after each statement.  x = 7 + 3 \* 6 / 2 – 1;  x = 2 % 2 + 2 \* 2 – 2 / 2;  x = ( 3 \* 9 \* ( 3 + ( 9 \* 3 / (3) ) ) ); |
|  | WAP to demonstrate if…else statement. |
|  | Write an application that declares 5 integers, determines and prints the largest and smallest in the group. |
|  | Write an application that declares 5 integers, calculates and print the average of these numbers |
|  | Write an application that declares two integers, determines whether the first is a multiple of the second and print the result. [Hint: Use the remainder operator.] |
|  | Write an application that calculates the squares and cubes of the numbers from 0 to 10 and prints the resulting values in a table format, a shown below.  Number Square Cube  0 0 0  1 1 1  2 4 8 and so on |
|  | Write an application that calculates the product of the odd integers from 1 to 15. |
|  | Write an application that evaluates the factorial of the integers from 1 to 5. |
|  | Write an application for calculating Compound-interest (interest rate of 5% for 10 years) with *for* loop. |
|  | Modify the above compound-interest application to repeat its steps for interest rates of 5, 6, 7, 8, 9 and 10%. Use a for loop to vary the interest rate. |
|  | A mail-order house sells five products whose retail prices are as follows : Product 1 : Rs. 99.90 , Product 2 : Rs. 20.20 , Product 3 : Rs. 6.87 , Product 4 : Rs. 45.50 and Product 5 : Rs. 40.49 . Write an application that reads a series of pairs of numbers as follows :  a) product number b) quantity sold  Your program uses a *switch* statement to determine the retail price for each product. It should calculate and display the total retail value of all products sold. |
|  | write application that take a series of integers in pair(e.g. (a,b)) and find the following  i) Is first one is greater  ii) Is both are even  iii) They are relatively prime or not |
|  | Write a Java program that randomly fill a 3 by 4 by 6 arrays and then prints the largest and smallest values in the array. |
|  | WAP in java to demonstrate String class operations. |
|  | Write an application that uses String method *equals* and *equalsIgnoreCase* to tests any two string objects for equality. |
|  | Write an application that uses *String* method *indexOf* to determine the total number of occurrences of any given alphabet in a defined text. |
|  | Write an application that uses *String* method *concat* to concatenate two defined strings. |
|  | Write an application that finds the length of a given string. |
|  | Write an application that uses *String* method *charAt* to reverse the string. |
|  | Write an application that finds the substring from any given string using *substring* method and *startsWith* & *endsWith* methods. |
|  | Write an application that changes any given string with uppercase letters, displays it , changes it back to lowercase letters and displays it. |
|  | Create a class called *Employee* that includes three pieces of information as instance variables – a first name (type String), a last name (type String) and a monthly salary (double) |
|  | Create a constructor in above class to initialize the three instance variables. Provide a *get* method for each instance variable. |
|  | Write a test application named *EmployeeTest* that demonstrates class Employee’s capabilities. Create two employee objects and display each object’s yearly salary. |
|  | Give each employee a 10% raise and display each Employee’s yearly salary again. |
|  | Create a class *Account* with an instance variable *balance* (double). It should contain a constructor that initializes the *balance*, ensure that the initial balance is greater than 0.0. |
|  | Create two methods namely *credit* and *getBalance*. The first one adds the amount (passed as parameter) to balance and does not return any data. The second method allows clients (i.e. the other classes that use this class) to obtain the value of a particular *Account* object’s *balance*. |
|  | Create class *AccountTest* to create and manipulate an *Account* object. |
|  | Write another method *debit* in the above program that withdraws money from an Account. Ensure that the debit amount does not exceed the Account’s *balance*. In that case the *balance* should be left unchanged and the method should print a message indicating “Debit amount exceeded account balance”. Modify class *AccountTest* to test method debit. |
|  | Write an application that reads a five digit integer and determine whether it is a palindrome (digit that reads the same backward and forward eg. 12321, 45554 etc.). Display an error message, if the number is no5 five digits long and allow the user to enter a new value. |
|  | Write an application that reads three nonzero values entered by the user and determines and prints sum, product, average, smallest & largest of three. |
|  | Write an application that prompts the user for the radius of a circle and uses a method called circleArea to calculate the area of the circle. |
|  | Add another method in the above program circlePerimeter to calculate the perimeter of the circle. |
|  | Write an application to create a super class Employee with information first name & last name and methods getFirstName(), getLastName() derive the sub-classes ContractEmployee and RegularEmployee with the information about department, designation & method displayFullName() , getDepartment, getDesig() to print the salary and to set department name & designation of the corresponding sub-class objects respectively. |
|  | Derive sub-classes of ContractEmployee namely HourlyEmployee & WeeklyEmployee with information number of hours & wages per hour, number of weeks & wages per week respectively & method calculateWages() to calculate their monthly salary. Also override getDesig () method depending on the type of contract employee. |
|  | Write an application to create a super class Vehicle with information vehicle number,insurance number,color and methods getConsumption() and displayConsumption(). Derive the sub-classes  TwoWheeler and FourWheeler with method maintenance() and average() to print the maintenance  And average of vehicle. |
|  | Extend the above TwoWheeler class with methods getType() and getName() which gives the information about the type and the name of the company.Create sub-classes Geared and NonGeared with method average() to print the average of a geared and non-geared two wheeler. |
|  | Create a super class CommunityMember with the information of member i.e. name, address, contact, date\_of\_join, through methods getName (), getAddress (), getContact (), getDate\_of\_Join () and derive sub-classes Employee and Student with method Qualification () to print the related information with his/her qualification. |
|  | Create a super class Shape with methods getName() which gives the information about the type of the shape.derive its sub-classes TwoDim and ThreeDim with method area() and volume() respectively which prints the area and volume of a two-dimensional and three-dimensional shape. |
|  | Extend the class TwoDim with methods getLength(),getBreadth() which displays the length and breadth of two dimentional shapes.Derive sub-classes rectangle, rhombus with method getArea() and getPerimeter() to calculate the area and perimeter of this two dimensional shapes. . |
|  | Extend the class ThreeDim with methods getLength(),getBreadth(),getHeight() which displays the length , breadth and height of three dimentional shapes.Derive sub-classes cuboid,tetrahedron with method getArea() and getVolume() to calculate the area and volume of this threee dimensional shapes. . |
|  | Create a super class Student with methods getQual (), getFirstName(),getLastName(), getAddress(), getContat(), which gives basic details of student.derive sub-classes Faculty and Scholar with method salary(), Course() resp. which gives the additional information about the salary and course of faculty and scholar resp. . |
|  | Create an abstract class Shape which calculates the area and volume of 2-d and 3-d shapes with methods getArea and getVolume. Reuse this class to calculate the area and volume of square, circle, cube. |
|  | Create an abstract class Employee with methods getAmount() which displays the amount paid to employee. Reuse this class to calculate the amount to be paid to WeeklyEmployeed and HourlyEmployee according to no. of hours and total hours for HourlyEmployee and no. of weeks and total weeks for WeeklyEmployee. |
|  | Create an Interface payable with method getAmount ().Calculate the amount to be paid to Invoice and Employee by implementing Interface. |
|  | Create an Interface Vehicle with method getColor(),getNumber(),getConsumption(). Calculate the fuel consumed, name and color for TwoWheeler and FourWheeler by implementing interface Vehicle. |
|  | Create an Interface Fare with method getAmount() to get the amount paid for fare of travelling. Calculate the fare paid by bus and train implementing interface Fare. |
|  | Create an Interface StudentFee with method getAmount(), getFirstName(),getLastName(), getAddress(), getContact(). Calculate the amount paid by the Hostler and NonHostler student by implementing interface StudentFee |
|  | WAP to create your own package. Package should have more than two classes. Write a class that uses the package. |
|  | WAP in java to sort array of strings |
|  | WAP in java to implement Selection Sort Algo. |
|  | WAP in java to implement Bubble Sort Algo. |
|  | WAP in java to implement Binary Search Algo. |
|  | WAP in java to demonstrate Stack class. |
|  | Exception Handling program for *IOException--actually contained in java.io*, but it is thrown if the JVM failed to open an I/O stream |
|  | Write a program that shows that the order of the catch blocks is important. If you try to catch a superclass exception type before a subclass type, the compiler should generate errors. |
|  | Program for demonstrating the use of throw, throws & finally - Create a class with a main( ) that throws an object of class Exception inside a try block. Give the constructor for Exception a String argument. Catch the exception inside a catch clause and print the String argument. Add a finally clause and print a message to prove you were there. |
|  | Create your own exception class using the extends keyword. Write a constructor for this class that takes a String argument and stores it inside the object with a String reference. Write a method that prints out the stored String. Create a try-catch clause to exercise your new exception. |
|  | Write a program to rethrow an exception – Define methods one () & two (). Method two () should initially throw an exception. Method one() should call two(), catch the exception and rethrow it Call one() from main() and catch the rethrown exception. |
|  | Write a program to change the priority of thread. |
|  | WAP for producer consumer problem (with synchronization). |
|  | Write an application that displays any string. Choose color from combo box to change the color of this displayed string. |
|  | WAP to demonstrate AWT buttons with event handling. |
|  | WAP to demonstrate BorderLayout in AWT window. |
|  | WAP in java to read a file using FileReader and break the contents using StringTokenizer. |
|  | WAP in java to write into a file using FileWriter. |
|  | WAP in java to demonstrate RandomAccessFile. |
|  | WAP in java to rename a file. |
|  | WAP in java to reverse a string by word. |
|  | WAP in java to demonstrate StringTokenizer class. |
|  | WAP to create your own package in defined a class StringUtils in this package. |
|  | WAP in java to create thread that print counting by extending Thread class. |
|  | WAP in java to demonstrate current thread. |
|  | WAP in java to create a thread using Runnable interface. |
|  | WAP in java to determine whether a thread is alive or not. |
|  | WAP in java to demonstrate join () method of Thread class. |
|  | WAP in java to implement toString() method in your class to print objects. |
|  | WAP in java to using pow () function. |
|  | WAP to demonstrate if…else statement. |
|  | WAP in java to sort array of strings |
|  | WAP in java to implement Selection Sort Algo. |
|  | WAP in java to implement Bubble Sort Algo. |
|  | WAP in java to implement Binary Search Algo. |
|  | WAP in java to demonstrate Stack class. |
|  | WAP in java using thread to create Remainder |
|  | WAP to demonstrate java.lang.reflact package. |
|  | WAP to determine whether an entered number is prime or not. |
|  | WAP to serialize an object and WAP to deserialize an object. |
|  | WAP to change background color according to selected color from combo box. |
|  | Write a program in java to scroll a string using Applet. |
|  | WAP in java to demonstrate all mouse events. |
|  | WAP in java to demonstrate all keyboard events. |
|  | WAP in java to demonstrate GridLayout. |
|  | WAP in java to demonstrate Check Boxes in applet. |
|  | WAP in java to demonstrate Calender class. |
|  | WAP in java to display all available fonts in an applet window |
|  | WAP to store student objects in a file. |
|  | WAP to display specific student record based on rollnumber. If not found than display all records in tabular form. |
|  | WAP to copy the content of a file to another file. |
|  | WAP in java to demonstrate Random class. |
|  | WAP in java to create an Applet that’s background color will be change on each second. |
|  | WAP in java to sum two 2-D matrixes and store the result into third matrix. |
|  | Create a program that will print every argument given on the command line. Consider how your program will deal with no argument. |
|  | WAP to draw a string and choose its size respectively from combo box. |
|  | WAP in java to demonstrate data entry program. |
|  | Create an application of cash withdrawal from the bank account that have no. of users that are operating the accounts.( synchronization) |
|  | WAP to create three text boxes and save entered value into a file. |
|  | Implement a class Reader that count the number of times a particular character, such as e, is read. The character can be specified when the stream is created. |
|  | Construct a program Wc ("word count"), which counts number of chars, words and lines of the text file. Space is counted as a character. Empty rows are counted as lines. "Word" will represent a string. |
|  | Write a small application with a default date 01/01/2000 and three combo boxes displaying valid days, months & year (1990 – 2050). Change the displayed date with the one chosen by user from these combo boxes. |
|  | Create a GUI with a text field and three buttons. When you press each button, make some different text appear in the text field. |
|  | Create a GUI application to take input of two numbers (text field) from user. When you press button it should display sum of the two numbers in a third text box. |
|  | Create an pro with a Button and a TextField. Write a referenceEvent( ) so that if the button has the focus, characters typed into it will appear in the TextField. |
|  | Write an application to create a GUI with two buttons such that clicking on the first displays the message “Welcome to SCS” on the window and clicking on the second changes the color of the message(*hint : toggle the color*) |
|  | Create a GUI with title STUDENT which has labels roll no., name, class, address with textboxes for taking input from the user(without any functionality). |
|  | Create a GUI application for fees receipt which contains checkboxes for selecting the course, radio buttons for selecting gender and labels and corresponding textboxes for name, class, date and amount paid. |
|  | Create a GUI application to display a calculator using grid Layout (You do not have to provide functionality). |
|  | WAP for string tokenizer. |
|  | Convert the input date in words. Input format is dd mm yy. |