

# GUJARAT TECHNOLOGICAL UNIVERSITY

## MASTER OF COMPUTER APPLICATION

### SEMESTER: III

Subject Name: **Programming Skills-IV (Java/ Statistical Methods)**

Subject Code: **630006**

---

#### **Suggested Practicals for Java and Statistical Methods:**

1. Observe the interactions involved in the process of booking a railway ticket. Identify the various objects involved and the interactions between the objects in order to solve a problem of booking a railway ticket.
2. Write a simple java application to print a pyramid with 5 lines. The first line has one character, 2nd line has two characters and so on. The character to be used in the pyramid is taken as a command line argument.
3. Write a Java application which takes several command line arguments, which are supposed to be names of students and prints output as given below:  
(Suppose we enter 3 names then output should be as follows):

Number of arguments = 3

1: First Student Name is =Tom

2: Second Student Name is =Dick

3: Third Student Name is =Harry

Hint: An array may be used for converting from numeric values from 1 to 20 into String.

4. Write a class, with main method, which declares floating point variables and observe the output of dividing the floating point values by a 0, also observe the effect of assigning a high integer value (8 digits and above) to a float and casting it back to int and printing.
5. Write a class called Statistics, which has a static method called average, which takes a one-dimensional array for double type, as parameter, and prints the average for the values in the array. Now write a class with the main method, which creates a two-dimensional array for the four weeks of a month, containing minimum temperatures for the days of the week(an array of 4 by 7), and uses the average method of the Statistics class to compute and print the average temperatures for the four weeks.
6. Define a class called Product, each product has a name, a product code and manufacturer name. Define variables, methods and constructors, for the Product class. Write a class called Test Product, with the main method to test the methods and constructors of the Product class.
7. Define a class called Cartesian Point, which has two instance variables, x and y. Provide the methods get X() and get Y() to return the values of the x and y values respectively, a method called move() which would take two integers as parameters and change the values of x and y respectively, a method called display() which would display the current values of x and y. Now overload the method move() to work with single parameter, which would set both x and y to the same values, .

Provide constructors with two parameters and overload to work with one parameter as well. Now define a class called Test Cartesian Point, with the main method to test the various methods in the Cartesian Point class.

8. Define a class called Triangle, which has constructor with three parameters, which are of type Cartesian Point, defined in the exercise 7. Provide methods to find the area and the perimeter of the Triangle, a method display() to display the three Cartesian Points separated by ':' character, a method move() to move the first Cartesian Point to the specified x, y location, the move should take care of relatively moving the other points as well, a method called rotate, which takes two arguments, one is the Cartesian Point and other is the angle in clockwise direction. Overload the move method to work with Cartesian Point as a parameter. Now define a class called Test Triangle to test the various methods defined in the Triangle class. Similarly also define a class called Rectangle which has four Cartesian Point.
9. Override the toString, equals and the hashCode methods of the classes Triangle and Rectangle defined in exercises 7 and 8 above, in appropriate manner, and also redefine the display methods to use the toString method.
10. Define an abstract class called Polygon. Provide a constructor which takes an array of Cartesian Point as parameter. Also provide method called perimeter, which calculates and returns the perimeter of the Polygon. Declare abstract method area for this class. Also define a method called move, which takes two parameters x and y to specify the destination for the first point of the Polygon, and overload to make it work for Cartesian Point as a parameter. Now update the classes Triangle and Rectangle in the exercise 8 above, to be a subclass of the Polygon class. Write appropriate class with main method to test the polymorphism in the area method.
11. Make the class Cartesian Point, belong to a package called edu. gtu. geometry, the classes Polygon, Triangle and Rectangle belong to the package edu. gtu. geometry. shapes and the classes Test Cartesian Point, Test Triangle, Test Rectangle and Test Polygon belong to the package edu. gtu. test. Use appropriate access specifiers for the classes and the members of the classes defined in the earlier exercises. Now onwards all the classes must be defined in a package.
12. Update the classes Triangle and Rectangle, to throw an exception if the Cartesian Point instances passed as parameter does not specify an appropriate Triangle or Rectangle. e.g. In case of Triangle, if the three points are in a straight line, or in case of Rectangle, if the lines when connected cross each other.
13. Define a class called Polygon Manager, which manages a number of Polygon instances. Provide methods to add, remove and list the Polygon instances managed by it. Test the methods of Polygon Manager by writing appropriate class with main method.
14. Define a class called Statistical Data which manages a number of readings of type int. Provide a method to set Data available in the form of an array, a method add to individually add data of type int, a method to reset the entire data, and methods to return the following statistics:
  1. mean
  2. median
  3. mode
  4. variance
  5. standard deviation
  6. specified percentile (between 0 – 100) the method should have a parameter.

7. specified quartile (between 1 - 3) the method should have a parameter.
8. interquartile range

Use this class to calculate the given statistics for the data regarding the total of marks obtained by all the students in your class in the previous exams. The data may be accepted one by one for each student from the keyboard.

15. Update the class Statistical Data, and define a method called load From CSV, which takes as parameter an Input Stream, where numeric data is available in an ASCII format, in a comma separated form. Overload this method to take a File instance as parameter. Test the new methods using appropriate data.
16. A college maintains the information about the marks of the students of a class in a text file with fixed record length. Each line in the file contains data of one student. The first 25 characters have the name of the student, next 12 characters have marks in the four subjects, each subject has 3 characters. Create a class called Student Marks, which has student Name, and marks for four subjects. Provide appropriate getter methods and constructors, for this class. Write an application class to load the file into an array of Student Marks. Use the Statistical Data class to compute the statistics mean, median, mode, standard deviation and interquartile range for each of the subjects in the class.
17. In the above exercise, use multithreading, to compute the statistics, after loading the Student Marks from the file, for marks information available for different classes available from files placed in a directory. Create at least five files in a directory with fixed record length to test your code.
18. In the class Statistics defined earlier add method to find the percentile for a given data value. Use this to find your percentile ranking your class, during the last exam.
19. A Frequency Distribution is a tabular summary of data showing the number (frequency) of items in each of several nonoverlapping classes. These classes may represent qualitative or quantitative data. Create a class called Frequency Distribution, which provides a method called load From File, to load data Strings from a file, and compute the Frequency Distribution for the various kinds of Strings available from the file. The class should provide a method to print the frequency distribution, and another to print the percent frequency distribution in a tabular manner. Test this class by using the sample data available regarding purchases of best selling paperback business book available from a file containing the abbreviated names of the book separated by white spaces.
20. Create subclass of the Frequency Distribution class called Frequency Distribution Quantitative, which provides method called add Class Range, which takes two numbers as parameter. The class should maintain a number of class ranges, and should ensure that the class ranges are nonoverlapping. The load From File method should be overridden to compute frequency distribution of numeric data loaded from the file, it should discard the data which does not fall in any of the class ranges. Test this class by using the sample data available regarding the waiting times of patient arriving at a clinic of a doctor for emergency service.
21. Update the FrequencyDistributionQuantitative class defined in exercise 19, and adds a method to reset the data and the class ranges. Also add a method called set Range Interval, which specifies the size of each class Range and the starting value, this should be used at building the nonoverlapping class ranges as the data is loaded, now only the data less than the starting value may be discarded while loading numeric data from the file. Test this class by using the data regarding the salary of marketing executives available in a file. Set the starting value as 5,000 and

have intervals of 500.

22. Create a class called Stem and Leaf, which should have a provision to maintain the stem size and the number of intervals between the two stems. It should also have a method to load the data from the file and a method to display the stem-and-leaf diagram for the loaded data. Test the class to display data regarding the marks of students of your class in the previous exam.
23. Create a class called Pie Canvas, which subclasses from the java. awt. Canvas class, and overrides the paint method to display the Pie Chart for the data available from an instance of Frequency Distribution. Provide a method to set the instance of Frequency Distribution, on the Canvas. Test this class by adding it into a Frame and using the Frequency Distribution instance created for data regarding the top selling books prepared for exercise 18.
24. Create a class called Statistical Data, which has capability of maintaining data regarding multiple variables. It should have a method to specify the variable names as String array and the method to load values from a file regarding the variables. eg. We consider two variables as percentage of marks in GCET exam and percentage of marks in 1<sup>st</sup> year of MCA, Provide methods in the class to compute the correlation coefficient between any two variables, specified in the parameter. Test the class by computing the correlation coefficient between the marks of GCET and marks of 1<sup>st</sup> year MCA for students of your class.
25. Simple random sampling uses a sample of size n from a population of size N to obtain data that can be used to make inferences about the characteristics of a population. Suppose that, from a population of 50 bank accounts, we want to take a random sample of four accounts in order to learn about the population. How many different random samples of four accounts are possible. Write a Java class which can compute the number of random samples for size n from a population of N, Also provide method to display all possible samples, and also a method to return a sample, derived from the possible samples. (Hint use random method of the Math class to select the random sample).
26. Dasher is an information-efficient text-entry interface, driven by natural continuous pointing gestures. Dasher is a competitive text-entry system wherever a full-size keyboard cannot be used. Dasher is a zooming interface. You point where you want to go, and the display zooms in wherever you point. The world into which you are zooming is painted with letters, so that any point you zoom in on corresponds to a piece of text. The more you zoom in, the longer the piece of text you have written. You choose what you write by choosing where to zoom. To make the interface efficient, we use the predictions of a language model to determine how much of the world is devoted to each piece of text. Probable pieces of text are given more space, so they are quick and easy to select. Improbable pieces of text (for example, text with spelling mistakes) are given less space, so they are harder to write. The language model learns all the time: if you use a novel word once, it is easier to write next time.

Write a class in Java which models the probabilities for the alphabets in a language as the starting alphabet and then the probabilities for the subsequent alphabets which may follow the alphabet and so on leading to the end of word. The probabilities may be built by reading a text file containing lots of texts in the given language. (Hint: create a class to represent a node, which has an alphabet and its probability and each node has capability of maintaining a list of subsequent nodes).  
(Advanced question)

27. Write a class in Java to plot a Histogram for the data available from the instance of Statistical Data class defined in exercise 24 earlier. Make a provision for selecting the variables for which

Histogram is to be plotted.

28. Write a class in Java to plot a Scatter diagram for the data available from the instance of Statistical Data class defined in exercise 24 earlier. Make a provision for selecting the variables for which scatter diagram is to be plotted.
29. Define a class which implements the Table Model for displaying the data available from the Statistical Data instance of exercise 24, in a tabular form. Use instance of this model to create an instance of J Table and display the data of exercise 24 in tabular format.
30. Create an applet which has a Text Field to accept a URL string, and displays the document of the URL string in a new browser window.