Exercise 1

Variables and Data Type

Exercise 1: Write a Java program to display Hello World on the screen.

Exercise 2:Write a Java program to display the asterisk pattern as shown below (Without using loop):

*****
*****
*****
*****

Exercise 3: Write a Java program to declare two integer variables, one float variable, and one string variable and assign 10, 12.5, and "Java programming" to them respectively. Then display their values on the screen.

Exercise 4: Write a Java program by using BufferedReader class to prompt a user to input his/her name and then the output will be shown as an example below:

Hello Dara!

JAVA ARITHMETIC OPERATORS

Exercise 1: Write Java program to allow the user to input two integer values and then the program prints the results of adding, subtracting, multiplying, and dividing among the two values.

See the example below:

Enter value a:30
Enter value b:10
The result of adding is 40.
The result of subtracting is 20;
The result of multiplying is 300.
The result of dividing is 3.

Exercise 2: Write Java program to generate a random number between 1 to 6.

To generate a random number, you can use the Random class of java.util package. You may use the abs() method of Math class to make sure you can get only a positive number.

COMPOUND OPERATORS

Exercise 1: Write Java program to allow the user to input two float values and then the program adds the two values together. The result will be assigned to the first variable.

Enter value a:12.5
The value of a before adding is 12.5.
Enter value b:34.9
The value of a after adding is 47.4.

Exercise 2: Write Java program to allow the user to input the amount of deposit, yearly interest rate (percentage), and income tax(percentage). Then the program will calculate the amount of interest that the person earns in the

year. See the example output below:
The amount of deposit: 1000
Yearly interest rate: 7.5%
Income tax rate: 4%
The amount of interest earned in the year:72.0
JAVA IF ELSE
Exercise 1: Write Java program to allow the user to input his/her age. Then the program will show if the person is eligible to vote. A person who is eligible to vote must be older than or equal to 18 years old.
Enter your age: 18
You are eligible to vote.
Exercise 2: Write a Java program to determine whether an input number is an even number.
IF ELSE AND LOGICAL OPERATORS
Exercise 1: Write a Java program that determines a student's grade.
The program will read three types of scores (quiz, mid-term, and final scores) and determine the grade based on the following rules:
-if the average score >=90% =>grade=A
-if the average score >= 70% and <90% => grade=B
-if the average score>=50% and <70% =>grade=C
-if the average score<50% =>grade=F
See the example output below:
Quiz score: 80
Mid-term score: 68
Final score: 90
Your grade is B.
Exercise 2: Write a Java program to calculate the revenue from a sale based on the unit price and quantity of a product input by the user.
The discount rate is 10% for the quantity purchased between 100 and 120 units, and 15% for the quantity purchased greater than 120 units. If the quantity purchased is less than 100 units, the discount rate is 0%. See the example output as shown below:
Enter unit price: 25
Enter quantity: 110
The revenue from sale: 2475.0$
After discount: 275.0$(10.0%)
SWITCH CASE
Exercise 1: Write a Java program to detect key presses.
If the user pressed number keys(from 0 to 9), the program will tell the number that is pressed, otherwise, program will show "Not allowed".
Exercise 2: Write a Java program that allows the user to choose the correct answer of a question.
See the example below:
What is the correct way to declare a variable to store an integer value in Java?

a. int 1x=10;
b. int x=10;
c. float x=10.0f;
d. string x="10";
Enter your choice: c
Invalid choice
a. int 1x=10;
b. int x=10;
c. float x=10.0f;
d. string x="10";
Enter your choice: b
Congratulations!

LOOPS IN JAVA

Exercise 1: Write a Java program by using two for loops to produce the output shown below:

```
*******
******
*****
****
***
**
*
```

Exercise 2: Write a Java program by using three for loops to print the following pattern:

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

WHILE LOOP

Exercise 1: Write Java program to prompt the user to choose the correct answer from a list of answer choices of a question.
The user can choose to continue answering the question or stop answering it.
See the example below:
What is the command keyword to exit a loop in Java?
a. int
b. continue
c. break
d. exit
Enter your choice: b
Incorrect!
Again? press y to continue:
Enter your choice: c

Congratulations!
Exercise 2:Write Java program to print the table of characters
that are equivalent to the Ascii codes from 1 to 122.
The program will print the 10 characters per line.
DO WHILE LOOP
Exercise 1: By using do while loop, write Java program to
prompt the user to choose the correct answer from a list of
answer choices of a question.
The user can choose to continue answering the question or stop answering it.
See the example below:
What is the command keyword to exit a loop in Java?
a. int
b. continue
c. break
d. exit
Enter your choice: b
Incorrect!
Again? press y to continue:
Enter your choice: c
Congratulations!
Exercise 2: By using do while loop, write Java program to
print the table of characters that are equivalent to the Ascii
codes from 1 to 122.
The program will print the 10 characters per line.

Exercise 2

Exercise-2
1. Read a sequence of characters from the terminal until q is read. Output
the largest character read, not
including the q.
2. Write a program that displays
the following pattern
1 2 3 4 5
2 4 6 8 10
3 6 9 12 15
4 8 12 16 20
5 10 15 20 25
6 12 18 24 30
7 14 21 28 35
8 16 24 32 40

Exercise 3

Exercise-3
1. Write a class that has three overloaded static methods for
calculating the areas of the following geometric shapes:

- circles
- rectangles
- cylinders

Here are the formulas for calculating the area of the shapes:
Area of a circle Area = pi*r^2
where: pi is Math.PI and r is the circle's radius
Area of a rectangle Area = Width x Length
Area of a cylinder Area = pi*r^2*h
where: n is Math.PI, r is the radius of the cylinder's base, and h is the cylinder's height
Because the three methods are to be overloaded, they should each have the same name, but different parameter lists. Demonstrate the class in a complete program.

2. Make a LandTract class that has two fields, once for the tract's length and one for the width. The class should have a method that returns the tract's area, as well as an equals method and a toString method. Demonstrate the class the class in a program that asks the user to enter the dimensions for the two tracts of land. The program should display the area of each and indicate whether the tracts are of equal size.

3. Write a class named month. This class should have an int field named monthNumber that holds the number of months. For example, January would be 1. Feburary would be 2, and so forth. In addition, provide this following methods.

- A no- arg constructor that sets the monthNumber field to 1.
- A constructor that accepts the number of the month as an argument. It should set the monthNumber field to the value passed as the argument. If a value less than 1 or greater than 12 is passed, the constructor should set monthNumber to 1.
- A Constructor that accepts the name of the month, such as "Janurary" or "February" as an argument. It should set the monthNUmber field to the correct corresponding value
- A setMonthNumber method that accepts an int argument, which is assigned to the monthNumber field. If a value less than 1 or greater than 12 is passed, the method should set monthNumber or 1.

・A getmonthNumber method that returns the value in the monthNumber field.

・A getMonthname method that returns the name of the month. For example, if the monthNumber field contains 1, then this method should return "january".

・A toString method that returns the same value as the getMonthName method.

・An equals method that accepts a month object as an argument. If the argument object holds the same data as the calling object, this method should return true. Otherwise it should return false.

・A greaterThan method that accepts a month object as an argument. If the calling object'smonthNumber field is greater than the argument's monthNumber field, this method should return true. Otherwise, it should return false.

・A lessThan method that accepts a Month object as an argument, If the calling object's monthNumber field is less than the argument's monthNumber field, this method should return true. Otherwise, it should return false.

4. For this assignment you will design a set of classes that work together to simulate a police officer issuing a parking ticket. You should design the following classes:

・TheParkedCarClass: This class should simulate a parked car. The class’s responsibilities are as follows:
- To know the car’s make, model, color, license number, and the number of minutes that the car has been parked.

・TheParkingMeterClass: This class should simulate a parking meter. The class’s only responsibility is as follows:
- To know the number of minutes of parking time that has been purchased.

・TheParkingTicketClass: This class should simulate a parking ticket. The class’s responsibilities are as follows:
-To report the make, model, color, and license number of the illegally parked car.

-To report the amount of the fine, which is $25 for the first hour or part of an hour that the car is illegally parked, plus $10 for every additional hour or part of an hour that the car is illegally parked
-To report the name and badge number of the police officer issuing the ticket.

• ThePoliceOfficerClass: This class should simulate a police officer inspecting parked cars. The class's responsibilities are as follows:
-To know the police officer's name and badge number.
-To examine a ParkedCar object and a ParkingMeter object, and determine whether the car's time has expired.
-To issue a parking ticket (generate a ParkingTicket object). if the car's time has expired
Write a program that demonstrates how these classes collaborate.

5. For this assignment, you will design a set of classes that work together to simulate a car's fuel gauge and odometer. The classes you will design are the following:
The FuelGauge class: This class will simulate a fuel gauge. its responsibilities are as follows:

• To know the car's current amount of fuel, in gallons.

• To report the car's current amount of fuel, in gallons.

• To be able to increment the amount of fuel by 1 gallon. This simulates putting fuel in the car. (The car can hold a maximum of 15 gallons.)

• To be able to decrement the amount of fuel by 1 gallon, if the amount of fuel is greater than 0 gallons. This simulates burning fuel as the car runs.

The odometer class: This class will simulate the car's odometer. its responsibilities are as follows:

• To know the car's current mileage.

• To report the car's current mileage.

• To be able to increment the current mileage by 1 mile. The maximum mileage the odometer can store is 999,999 miles. When this amount is exceeded, the odometer resets the current mileage to 0.

• To be able to work with a FuelGauge object. It should decrease the FuelGauge object's current amount of fuel by 1 gallon for every 24 miles traveled. (the car's fuel

economy is 24 miles per gallon.)
Demonstrate the classes by creating instances of each.
Simulate filling the car up with fuel, and then run a loop that increments the odometer until the car runs out of fuel.
During each loop iteration, print the car's current mileage and amount of fuel.
6. Design a Geometry class with the following methods:

- A static method that accepts the radius of a circle and returns the area of the circle.

Use the following formula:
Area=πr^2
Use Math.PI for π and the radius of the circle for r.

- A static method that accepts the length and width of a rectangle and returns the area of the rectangle. Use the following formula:

Area = Length x Width

- A static method that accepts the length of a triangle's base and the triangle's height. The method should return the area of the triangle. Use the following formula:

Area = Base x Height x 0.5
The methods should display and error message if negative values are used for the circle's radius, the rectangle's length or width, or the triangle's base or height.
Next write a program to test the class, which displays the following menu and responds to the user's selection:
Geometry Calculator
1. Calculate the Area of a Circle
2. Calculate the Area of a Rectangle
3. Calculate the Area of a Triangle
4. Quit
Entre your choice (1 - 4):
Display an error message if the user enters a number outside the range of 1 through 4 when selection an item from the menu.
7.
8.
Modify the program above to incorporate a constructor to provide initial value.
9.
10.Modify the program above to include constructors for all the three classes.
11.
12. Write a class named employee with the following fields:
name: The name field references a String object that holds the employee's name.
idNumber: The idNumber is an int variable that holds the employee's ID number.

department: The department field references a String object that holds the name of the department where the employee works.
position: The position field references a String object that holds the employee's job title.
The class should have the following constructors:
-A constructor that accepts the following values as arguments and assigns them to the appropriate fields: employee's name, employee's ID number, department, and position.
-A constructor that accepts the following values as arguments and assigns them to the appropriate fields: employee's name and ID number. The department and position fields should be assigned an empty string(“ ”).
-A no-arg constructor that assigns empty strings (“ ”) to the name, department, and position fields, and 0 to the idNumber field.
Write appropriate mutator that store values in these fields and accessor methods that return the values in those fields. Once you have the class, write a separate program that creates three Employee objects to hold the following data:
Name Id Number Department Position
Susan Meyers 47899 Accounting vice Pres.
Mark Jones Joy 39119 IT Programmer
Rogers 81774 Manufacturing Engineer
The program should store this data in the three objects and then display the data for each employee on the screen.
13.Design a class that holds the following personal data: name, address, age, and phone
number. Write appropriate accessor and mutator functions. Demonstrate the class by writing a program that creates three instances of it. One instance should hold your information, and the other two should hold your friends or family members information.
14.Design a Payroll class that has fields for an employee's name, ID number, hourly pay rate, and number of hours worked. Write the appropriate accessor and mutator methods and a constructor that accepts the employee's name and ID number as arguments. The class should also have a method that returns the employee's gross pay, which is calculated as the number of hours worked multiplied by the hourly rate. Write a program that demonstrates the class by creating a Payroll object, then asking the user to enter the data for an employee. The program should display the amount of gross pay earned.
15.Write a class named RetailItem that holds data about an item in a retail store. The class should have the following fields:
Description: The description field references a String object that holds a brief description of the item.
unitsOnHand: The unitsOnHand field is an int variable that holds the numbers of units currently in inventory.
Price: The price filed is a double that holds the item's retail price.
Write a constructor that accepts arguments for each field, appropriate mutator methods that store values in these fields, and accessor methods that returns the values in these fields. Once you have written the class, write a separate program that creates three RetailItem objects and stores the following data in them:

Description Units on Hand Price
Item #1 Jacket 12 59.95
Item #2 Designer Jeans 40 34.95
Item #3 Shirt 20 24.95

16. Design a TestScores class that has fields to hold three test scores. The class should have a constructor, accessor and mutator methods for the test score fields, and a method that returns the average of the test scores. Demonstrate the class by writing a separate program that creates an instance of the class. The program should ask the user to enter three test scores, which are stored in the TestScores object. Then the program should display the average of the scores, as reported by the TestScores object.

17. Write a Circle class that has the following fields:
radius: a double
PI: a final double initialized with the value 3.14159
The class should have the following methods:
Constructor: accepts the radius of the circle as an argument.
Constructor: A no-arg constructor that sets the radius field to 0.0.
setRadius: A mutator method for the radius field.
getRadius: An accessor method for the radius field.
getArea: Returns the area of the circle which is calculated as area=PI*radius*radius
getDiameter: Returns the diameter of the circle which is calculated as diameter=radius*2
getCircumference: Returns the circumference of the circle, which is calculated as circumference= 2*PI*radius
Write a program that demonstrates the Circle class by asking the user for the circle's radius, creating a Circle object, and then reporting the circle's area, diameter, and circumference.

18. A bank charges $10 per month plus the following check fees for a commercial checking
account:
$.10 each for fewer than 20 checks
$.08 each for 20 39 checks
$.06 each for 40 59 checks
$.04 each for 60 or more checks
The bank also charges an extra $15 if the balance of the account falls below $400
(before any check fees are applied). Write a program that asks for the beginning balance
and the number of checks written. Compute and display the bank s service fees
for the month.

19. Design a SavingsAccount class that stores a savings account’s annual interest rate and
balance. The class constructor should accept the amount of the savings account’s starting

balance. The class should also have methods for subtracting the amount of a withdrawal,
adding the amount of a deposit, and adding the amount of monthly interest to the balance.
The monthly interest rate is the annual interest rate divided by twelve. To add the monthly interest to the balance, multiply the monthly interest rate by the balance, and add
the result to the balance.
Test the class in a program that calculates the balance of a savings account at the end of a period of time. It should ask the user for the annual interest rate, the starting balance, and the number of months that have passed since the account was established. A loop should then iterate once for every month, performing the following:
2. Ask the user for the amount deposited into the account during the month. Use the class
method to add this amount to the account balance.
3. Ask the user for the amount withdrawn from the account during the month. Use the class
method to subtract this amount from the account balance.
4. Use the class method to calculate the monthly interest.
After the last iteration, the program should display the ending balance, the total amount of deposits, the total amount of withdrawals, and the total interest earned.
20.The following table lists the freezing and boiling points of several substances.
Substance Freezing Point Boiling Point
Ethyl Alcohol -173 172
Oxygen -362 -306
Water 32 212
Design a class that stores a temperature in a temperature field and has the appropriate accessor andmutator methods for the field. In addition to appropriate constructors, the class should have the following methods:

- isEthylFreezing: This method should return the boolean value true if the temperature stored in the temperature field is at or below the freezing point of ethyl alcohol. Otherwise, the method should return false.
- isEthylBoiling: This method should return the boolean value true if the temperature stored in the temperature field is at or above the boiling point of ethyl alcohol. Otherwise,
the method should return false.
- isOxygenFreezing: This method should return the boolean value true if the temperature
stored in the temperature field is at or below the freezing point of oxygen. Otherwise, the
method should return false.
- isOxygenBoiling: This method should return the boolean value true if the temperature

stored in temperature field is at or above the boiling point of oxygen. Otherwise, the method should return false.

- isWaterFreezing: This method should return the boolean value true if the temperature stored in the temperature field is at or below the freezing point of water. Otherwise, the method should return false.
- isWaterBoiling: This method should return the boolean value true if the temperature stored in the temperature field is at or above the boiling point of water. Otherwise, the method should return false.

Write a program that demonstrates the class. The program should ask the user to enter a temperature, and then display a list of the substance that will freeze at that temperature and those that will boil at that temperature. For example, if the temperature is
-20 the class should report that water will freeze and oxygen will boil at that temperature.

21.Write a class named Coin. The Coin class should have the following field:

- A String named sideUp. The sideUp field will hold either “heads” or “tails” indicating the side of the coin that is facing up.

The Coin class should have the following methods:

- A no-arg constructor that randomly determines the side of the coin that is facing up (“heads” or “tails”) and initializes the sideUp field accordingly.
- A void method named toss that simulates the tossing of the coin. When the toss method is called, it randomly determines the side of the coin that is facing up (“heads” or “tails”) and sets the sideUp field accordingly.
- A method named getSideUp that returns the value of the sideUp field.

Write a program that demonstrates the Coin class. The program should create an instance
of the class and display the side that is initially facing up. Then, use a loop to toss the coin 20 times. Each time the coin it tossed, display the side that is facing up. The program
should keep count of the number of times heads is facing up and the number of times tails
is facing up, and display those values after the loop finishes.

Exercise 4

Exercise-4

Q1. You are required to design a class hierarchy supporting the polymorphic code reuse for printing the details of a student that varies with the type of student i.e. whether he is UG, PG, PhD or some other type of student added in future. You should clearly bring out the differences among the design of class hierarchy, application code where polymorphic code reuse is supported and client code which exploits reuse supported through above design. Explain various advantages obtained through above design. Implement above design using Java/C++. (Student details for UG may be student name, roll-no,

department, SPI, CPI, …, For PG some fields like UG degree, UG institution may be added, further for PhD the name of supervisor(s) may be added.)

Exercise 5

Exercise-5
Arrays
1. Rainfall Class
Write a RainFall class that stores the total rainfall for each of 12 months into an array of doubles. The program should have methods that return the following;
• the total rainfall for the year
• the average monthly rainfall
• the month with the max rain
• the month with the least rain
Demonstrate the class in a complete program.
2. Payroll Class
Write a Payroll class that uses the following arrays as fields:
• employeeId. An array of seven integers to hold employee identification numbers. The array should be initialized with the following numbers:
5658845 4520125 7895122 8777541
8451277 1302850 7580489
• hours. An array of seven integers to hold the number of hours worked by each employee
• payRate. An array of seven doubles to hold each employees hourly pay rate
• wages. An array of seven doubles to hold each employees gross wages
The class should relate the data in each array through the subscripts. For example, the number in element 0 of the hours array should be the number of hours worked by the employee whose identification number is stored in element 0 of the employeeId array. That same employee's pay rate should be stored in element 0 of the payRate array.
In addition to the appropriate accessor and mutator methods, the class should have a method that accepts an employee s identification number as an argument and returns the gross pay for that employee.
Demonstrate the class in a complete program that displays each employee number and asks the user to enter that employee's hours and pay rate. It should then display each employee's identification number and gross wages.
3. Charge Account Validation
Create a class with a method that accepts a charge account number as its argument. The method should determine whether the number is valid by comparing it to the following list of valid charge account numbers:
5658845 4520125 7895122 8777541 8451277 1302850
8080152 4562555 5552012 5050552 7825877 1250255
L00523L 6545231 3852085 7576651 7881200 4581002
These numbers should be stored in an array or an ArrayList object. Use a sequential search to locate the number passed as an argument. If the number is in the array, the method should return true, indicating the number is valid. If the

number is not in the array, the method should return false, indicating the number is invalid.
Problem Write a program that tests the class by asking the user to enter a charge account number.
the program should display a message indicating whether the number is valid or invalid.
4. Charge Account Modification
Modify the charge account validation class that you wrote for Programming Challenge 3 so it reads the list of valid charge account numbers from a file. Use Notepad or another text editor to create the file.
5. Driver's License Exam
The local Driver's License Office has asked you to write a program that grades the written portion of the driver's license exam. The exam has 20 multiple choice questions. Here are the correct answers:
1. B 6. A 11. B 16. C
2. D 7. B 12. C 17. C
3. A 8. A 13. D 18. B
4. A 9. C 14. A 19. D
5. C 10. D 15. D 20. A
A student must correctly answer 15 of the 20 questions to pass the exam.
Write a class named DriverExam that holds the correct answers to the exam in an array field.
The class should also have an array field that holds the student's answers. The class should have the following methods:
• passed. Returns true if the student passed the exam, or false if the student failed
• totalCorrect. Returns the total number of correctly answered questions
• totallncorrect. Returns the total number of incorrectly answered questions
• questionsMissed. An int array containing the question numbers of the questions that
the student missed
Demonstrate the class in a complete program that asks the user to enter a student's answers, and then displays the results returned from the DriverExam class's methods.
6. Quarterly Sales Statistics
Write a program that lets the user enter four quarterly sales figures for six divisions of a company. The figures should be stored in a two-dimensional array. Once the figures are entered, the program should display the following data for each quarter:
• A list of the sales figures by division
• Each division's increase or decrease from the previous quarter (this will not be displayed for the first quarter)
• The total sales for the quarter
• The company's increase or decrease from the previous quarter (this will not be displayed for the first quarter)
• The average sales for all divisions that quarter
• The division with the highest sales for that quarter

7. Grade Book

A teacher has five students who have taken four tests. The teacher uses the following grading
scale to assign a letter grade to a student, based on the average of his or her four test scores:

Test Score	Letter Grade
90-100	A
80-89	B
70-79	C
60-69	D
0-59	F

Write a class that uses a string array or an ArrayList object to hold the five student's names, an array of five characters to hold the five students' letter grades, and five arrays of four doubles each to hold each student's set of test scores. The class should have methods that return a specific student's name, the average test score, and a letter grade based on the average.

Demonstrate the class in a program that allows the user to enter each student's name and his or her four test scores. It should then display each student's average test score and letter grade.

8. Grade Book Modification

Modify the grade book application in Programming Challenge 7 so that it drops each student's lowest score when determining the test score averages and letter grades.

9. Lottery Application

Write a Lottery class that simulates a lottery. The class should have an array of five integers named lotteryNumbcrs. The constructor should use the Random class (from the Java API) to generate a random number in the range of 0 through 9 for each clement in the array. The class should also have a method that accepts an array of five integers that represent a person's lottery picks. The method is to compare the corresponding elements in the two arrays and return the number of digits that match. For example, the following shows the lotteryNumbers array and the user's array with sample numbers stored in each. There are two matching digits (elements 2 and 4).

lotteryNumbers array:
7 4 9 1 3
User's array :
4 2 9 7 3

In addition, the class should have a method that returns a copy of the lotteryNumbers array.

Demonstrate the class in a program that asks the user to enter five numbers. The program should display the number of digits that match the randomly generated lottery numbers. If all of the digits match, display a message proclaiming the user a grand prize winner.

10. Array Operations

Write a program with an array that is initialized with test data. Use any primitive data type of your choice. The program should also have the following methods:

• getTotal. This method should accept, a one-dimensional array as its argument and

return the total of the values in the array.
• getAverage. This method should accept a one-dimensional array as its argument and return the average of the values in the array.
• getHighest. This method should accept a one-dimensional array as its argument and return the highest value in the array.
• getLowest. This method should accept a one-dimensional array is us argument and return the lowest value in the array.
Demonstrate each of the methods in the program.
11. 2D Array Operations
Write a program that creates a two-dimensional array initialized with test data. Use any primitive data type that you wish. The program should have the following methods:
• getTotal. This method should accept a two-dimensional array as its argument and
return the total of all the values in the array.
• getAverage. This method should accept a two-dimensional array as its argument and
return the average of all the values in the array.
• getRowTotal. This method should accept a two-dimensional array as its first argument
and an integer as its second argument. The second argument should be the subscript
of a row1 in the array. The method should return the total of the values in the specified row.
• getColumnTotal. This method should accept a two-dimensional array as its first argument
and an integer as its second argument. The second argument should he the subscript
of a column in the array. The method should return the total of the values in the
specified column.
• getHighestInRow. This method should accept a two-dimensional array as its first
argument and an integer as us second argument. The second argument should be the
subscript of a row in the array. The method should return the highest value in the
specified row of the array,
• getLowestInRow, This method should accept a two-dimensional array as its first argument
and an integer as its second argument. The second argument should be the subscript of a row in the array. The method should return the lowest value in the specified row of the array.
Demonstrate each of the methods in this program.
12. Phone Book ArrayList
Write a class named PhoneBookEntry that has fields for a person's name and phone

number. The class should have a constructor and appropriate accessor and mutator methods. Then write a program that creates at least five PhoneBookEntry objects and stores them in an ArrayList. Use a loop to display the contents of each object in the ArrayList.

13. Trivia Game

In this programming challenge, you will create a simple trivia game for two players. The program will work like this;

• Starting with player 1, each player gets a turn at answering 5 trivia questions. (There are 10 questions, 5 for each player.) When a question is displayed, four possible answers are also displayed. Only one of the answers is correct, and if the player selects the correct answer, he or she earns a point.

• After answers have been selected for all of the questions, the program displays the number of points earned by each player and declares the player with the highest number of points the winner.

You are to design a Question class to hold the data for a trivia question. The Question class should have String fields for the following data:

• A trivia question
• Possible answer 1
• Possible answer 2
• Possible answer 3
• Possible answer *4*
• The number of the correct answer (1,2, 3, or 4)

The Question class should have appropriate constructors), accessor, and mutator methods. The program should create an array of 10 Question objects, one for each trivia question. (If you prefer, you can use an ArrayList instead of an array.) Make up your own trivia questions on the subject or subjects of your choice for the objects.

Exercise 6

Assignment - 6

1. Write a Java program to retrieve an element (at a specified index) from a given array list.
2. Write a Java program to update specific array element by given elemen
3. Write a Java program to empty an array list.
4. Write a Java program to trim the capacity of an array list the current list size.
5. Write a Java program to append the specified element to the end of a linked list.
6. Write a Java program to get the first and last occurrence of the specified elements in a linked list.
7. Write a Java program to check if a particular element exists in a linked list.
8. Write a Java program to get the number of elements in a hash set.
9. Write a Java program to compare two sets and retain elements which are same on both sets.
10. Write a Java program to add all the elements of a specified tree set to another tree set.
11. Write a Java program to clone a tree set list to another tree set.
12. Write a Java program to create a new priority queue, add some colors (string) and print out the elements of the priority queue.
13. Write a Java program to retrieve the first element of the priority queue.

14. Write a Java program to copy all of the mappings from the specified map to another map.
15. Write a Java program to get all keys from the given a Tree Map.
16. Write a Java program to get a key-value mapping associated with the greatest key less than or equal to the given key.
17. Write a Java program to get the greatest key strictly less than the given key. Return null if there is no such key.
18. Write a Java program to get the least key greater than or equal to the given key. Returns null if there is no such key.

Exercise 7

Exercise 7
1. WAP in java to demonstrate multiple threads using Thread class?
2. WAP in java to demonstrate multiple threads using Runnable interface?
3. WAP in java to demonstrate sleep(), join() and setPriority() methods?
4. WAP in java to create Multi-threaded Chat Application in Java?
5. WAP in java to build calculator so that it can handle all the exceptions?
6. WAP in java that count how many prime numbers between minimum and maximum values provided by user. If minimum value is greater than or equal to maximum value, the program should throw a Invalid Range exception and handle it to display a message to the user on the following format: Invalid range: minimum is greater than or equal to maximum. For example, if the user provided 10 as maximum and 20 as minimum, the message should be: Invalid range: 20 is greater than or equal to 10?

Exercise 8

Exercise – 8
Question: - Implement following design pattern in java
1. Singleton
2. factory
3. Observer
4. Visitor
5. Iterator
6. Adapter
7. Prototype
8. Proxy

Exercise 9

17th April - Today's Exercise
P1: Java Coordinate System
P2: Drawing a Face
P3: Add face to Web page
P1: The Java Coordinate System

The Java coordinate system : Under this system, the upper left-hand corner of the window is the point
(0,0). The X axis goes across the window, and the Y axis goes down the window. So the bigger the X
value, the farther a point is to the right. The bigger the Y value, the farther it is down. There are no
negative X or Y values in the Java coordinate system. Actually, you can use negative values, but since
they're off the screen they won't show up!

1. Save files Coords.java and Coords.html to your directory. File Coords.java contains an applet that
draws a rectangle whose upper lefthand corner is at 0,0. Use the applet viewer to run this applet.
Remember that you have to do it through the html file: appletviewer Coords.html.
2. Modify the applet so that instead of 0,0 for the upper lefthand corner, you use the coordinates of
the middle of the applet window. This applet is set up to be 600 pixels wide and 400 pixels high, so you
can figure out where the middle is. Save, compile, and view your applet. Does the rectangle appear to
be in the middle of the screen? Modify the coordinates so that it does appear to be in the middle.
3. Now add four more rectangles to the applet, one in each corner. Each rectangle should just touch
one corner of the center rectangle and should go exactly to the edges of the window.
4. Make each rectangle be a different color. To do this, use the setColor method of the Graphics
class to change the color (this is already done once). Do not change the background color once it has
been set! Doing so causes the screen to flicker between colors.

P2: Drawing a Face

Write an applet that draws a smiling face. Give the face eyes with pupils, ears, a nose, and a mouth.
Use at least three different colors, and fill in some of the features. Name this file Face.java, and create a
corresponding .html file. View your applet using the applet viewer.

P3: Add face to your web page

Now add your face to a web page you created (you may have created one in an earlier lab exercise).
You will do this using the <APPLET> tag that is in the .html file you are using with the applet viewer
—you can just copy it out of that file and paste it into your other file. The applet will appear wherever
you insert the applet tag.
Submit the following files:

· Coords.java (the final version with all changes)

· Coords.html
· Face.java
· Face.html
· Your webpage html file