Practice Exercise

This document provides a list of exercises to be practiced by learners. Please raise a feedback in Talent Next, should you have any queries.

|  |  |
| --- | --- |
| Skill | Java 8 |
| Document Type | Lab Practice Exercises |
| Author | L & D |
| Current Version | 1.0 |
| Current Version Date | 16-June-2021 |
| Status | Active |

Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Change Date | Change Description | Changed By |
| 1.0 | 16-June-2021 | Added new problems on Logic Building, Type Conversions, Flow Control Statements, Java Methods and Arrays, Inheritance, Abstract Class, and Interface, String Handling and Exception Handling and Inner classes, Collection Framework | Anup Kini |

Contents

[Practice Exercise 1](#_Toc74820343)

[Document Control 2](#_Toc74820344)

[Problem Statement 1: Logic Building 4](#_Toc74820345)

[Problem Statement 2: Working with Type Conversions, Flow Control Statements 5](#_Toc74820346)

[Problem Statement 3: Working with Java Methods and Arrays 6](#_Toc74820347)

[Problem Statement 4: Working with Class and Object 7](#_Toc74820348)

[Problem Statement 5: Working with Inheritance, Abstract Class, and Interface 9](#_Toc74820349)

[Problem Statement 6: Working with String Handling and Exception Handling and Inner Classes 10](#_Toc74820350)

[Problem Statement 7: Data Manipulations using Collections (List, Set and Map) 12](#_Toc74820351)

Note: Every Problem Statement start in a new page

Problem Statement 1: Logic Building

**Important Instructions:**

* Refer the LAG document for accessing the Java 8 online playground.
* Refer <https://www.studytonight.com/code/playground/java/> link to access online playground.

1. Given two integers m and n. The goal is simply to swap their values in the memory block and writing the java code demonstrating below approaches.
   * Creating an auxiliary memory cell in the memory.
   * Without creating any auxiliary(additional) memory cell
   * Using exclusive OR (Bitwise XOR) operator
2. Write a Java program to check if a given integer is odd or even.
3. Given three numbers x, y, and z of which aim is to get the largest among these three numbers.
4. For a given number N, the purpose is to find all the prime numbers from 1 to N.

Examples:

Input: N = 11

Output: 2, 3, 5, 7, 11

Input: N = 7

Output: 2, 3, 5, 7

1. Write a Java Program to find sum of fibonacci series numbers of first N even indexes
   * For a given positive integer N, the purpose is to find the value of F2 + F4 + F6 +………+ F2n till N number. Where Fi indicates the ith Fibonacci number.
   * The Fibonacci Series are the numbers in the below-given integer sequence.
   * Series : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ……

Example:

Input: n = 4

Output: 33

N = 4, So here the Fibonacci series will be produced from 0th term till 8th term:

0, 1, 1, 2, 3, 5, 8, 13, 21

Sum of numbers at even indexes = 0 + 1 + 3 + 8 + 21 = 33.

Problem Statement 2: Working with Type Conversions, Flow Control Statements

1. Given a decimal number as input, we need to write a program to convert the given decimal number into equivalent binary number.

Examples:

Input : 7

Output : 111

Input : 10

Output : 1010

Input: 33

Output: 100001

1. Given a character in Java, the task is to convert this character into an integer.

Examples:

Input: ch = '3'

Output: 3

Input: ch = '9'

Output: 9

1. Write a Java program to print pyramid number pattern as follows,

A pyramid number pattern of row size r = 5 would look like:

1

2 3 2

3 4 5 4 3

4 5 6 7 6 5 4

5 6 7 8 9 8 7 6 5

1. Write a Java program to print star pascal’s triangle as follows,

Input : N = 5

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

Problem Statement 3: Working with Java Methods and Arrays

1. Write a Java program to find factorial of a number using recursion.

* The factorial of a positive number n is given by factorial of n (n!) = 1 \* 2 \* 3 \* 4 \* ... \* n
* The factorial of a negative number doesn't exist. And the factorial of 0 is 1.

1. Write a Java program to calculate the power of number using recursion.

Example:

Input: base = 3 powerRaised = 4;

Output: 3^4 = 81

1. Write a Java program to find largest element of an array

Example:

Input: numArray = { 23.4, -34.5, 50.0, 33.5, 55.5, 43.7, 5.7, -66.5 }

Output: Largest Number is 55.50

1. Write a Java program to calculate average using arrays

Example:

Input: numArray = { 45.3, 67.5, -45.6, 20.34, 33.0, 45.6 }

Output: The average is: 27.69

1. Write a Java program to add two matrix using multi-dimensional arrays

Example:

Input:

firstMatrix = { {2, 3, 4}, {5, 2, 3} };

secondMatrix = { {-4, 5, 3}, {5, 6, 3} };

Output:

Sum of two matrices is:

-2 8 7

10 8 6

Problem Statement 4: Working with Class and Object

1. Write a Java program to add two complex numbers by passing objects to a function. Refer below code snippet and write the required logic into it.

public class Complex {

double real;

double imag;

public Complex(double real, double imag){// Write Code Here}

public static Complex add(Complex n1, Complex n2){// Write Code Here}

public static void main(String[] args) {

Complex n1 = new Complex(2.3, 4.5),

n2 = new Complex(3.4, 5.0),

temp;

temp = add(n1, n2);

System.out.printf("Sum = %.1f + %.1fi", temp.real, temp.imag);

}

}

1. Write a Java code to overload sum method in Java. Refer the below code snippet and complete the code,

public class Sum {

public int sum(int x, int y){ // Write Code Here }

public int sum(int x, int y, int z){ // Write Code Here }

public double sum(double x, double y) { // Write Code Here }

public static void main(String args[])

{

Sum s = new Sum();

System.out.println(s.sum(10, 20));

System.out.println(s.sum(10, 20, 30));

System.out.println(s.sum(10.5, 20.5));

}

}

1. Write a Point class containing following members:

* private int x;
* private int y;
* Constructors, Getters/Setters

Write a Line class containing following members:

* Data Members

private Point p1;

private Point p2;

* Constructors, Getters/Setters,
* Methods

public void accept();

public void print();

public int getLength(); // use distance formula

In main() method, create object of the Line class and test all functionalities provided in it.

Problem Statement 5: Working with Inheritance, Abstract Class, and Interface

Solve following sub problems,

1. Create two classes, Person and Address, with default constructors (empty argument lists) that announce themselves. Inherit a new class called Employee from Person and create a member of class Address inside Employee. Do not create a constructor for Employee. Create an object of class Employee and observe the results.
2. Modify problem 1 so that Person and Address have constructors with arguments instead of default constructors. Write a constructor for Employee and perform all initialization within Employee’s constructor.
3. Create an abstract class Instrument which is having the abstract function play. Create three more sub classes from Instrument which is Piano, Flute, Guitar. Override the play method inside all three classes printing a message as follows,

**“Piano is playing tan tan tan tan” for Piano class**

**“Flute is playing toot toot toot toot” for Flute class**

**“Guitar is playing tin tin tin” for Guitar class**

Note: You must not allow the user to declare an object of Instrument class.

* Create an array of 10 Instruments.
* Assign different type of instrument-to-Instrument reference.
* Check for the polymorphic behavior of play method.
* Use the instanceof operator to print that which object stored at which index of instrument array.

1. Create an interface MedicineInfo to represent a drug manufactured by a pharmaceutical company. Provide an abstract method displayLabel() in this interface to print Name and address of the company. Do following tasks,

* Implement MedicineInfo interface with Tablet, Syrup and Ointment classes.
* Override the displayLabel() method in each of these classes to print information suitable to the type of medicine. For example, in case of tablets, it could be “store in a cool dry place”, in case of ointments it could be “for external use only” etc.

Create a class TestMedicine . Write main function to do the following:

* Declare an array of Medicine references of size 10.
* Create a medicine object of the type as decided by a randomly generated integer in the range 1 to 3.(Refer Java API Documentation to find out random generation feature.)
* Check the polymorphic behavior of the displayLabel() method.

Problem Statement 6: Working with String Handling and Exception Handling and Inner Classes

1. Manipulate the command line arguments as given below. If a “Test” class is run as:

**java Test JAVA is Simple**

Perform the String manipulations such that the output should be following,

* 1. JAVA IS SIMPLE Hint : [replace(), toUpperCase()]
  2. java is simple Hint : [replace(), toLowerCase()]
  3. J i S Hint : [split(), charAt(0)]
  4. Simple is JAVA Hint : [split()]
  5. AVAJ si elpmiS Hint : [split(), StringBuilder.reverse()]
  6. Total length is: 12 Hint : [split(), length()]

1. Generate a 3-digit, 5-digit, and 10-digit random number by using Math.random(), Math.pow(x,y) and display it.
2. Create a class called CalcAverage that has the following method:

**public double avgFirstN(int N)**

* This method receives an integer as a parameter and calculates the average of first N natural numbers.
* If N is not a natural number, throw an exception IllegalArgumentException with an appropriate message.

1. Create a class Number having the following features:

* Data Members

int a

int b

result double => stores the result of arithmetic operations performed on a and b.

* Member functions

Number(x, y) => constructor to initialize the values of a and b

add() => stores the sum of a and b in result

sub() => stores difference of a and b in result

mul() => stores product in result

div() => stores a divided by b in result

* Test to see if b is 0 and throw an appropriate exception since division by zero is undefined.
* Display a menu to the user to perform the above four arithmetic operations.

1. Create a Robot class and add following functionality inside,

* Define a private data member int id
* Define a non-static nested class called Brain, which will have public void think() method. Try accessing id inside the think() method of nested class Brain.
* Define a static nested class called Battery, which will have public void charge() method. Display a simple message as “Battery is charging” inside charge() method.
* Define start() method inside the outer class Robot as follows

public void start() {

// Display message as "Starting robot with ID : 1234”

// Create Instance of Brain class and class think method

// Create Instance of outer class Robot and call static method charge() of Battery class.

}

* Create main() method and class start() method of Robot class.

Problem Statement 7: Data Manipulations using Collections (List, Set and Map)

Solve following sub problems,

* 1. Write a program to add list of student names to **ArrayList** and it should find a particular name whether it exists or not in the list.
  2. Create a Product class with Product Id & Product Name. Write a program to accept information of 10 products and store that in HashSet. Do following operations,

1. Search a particular product in the HashSet.
2. Remove a particular product from the HashSet by using product id.

(refer below table for the product list)

|  |  |
| --- | --- |
| **Product Id** | **Product Name** |
| P001 | Maruti 800 |
| P002 | Maruti Zen |
| P003 | Maruti Dezire |
| P004 | Maruti Alto |

* 1. Implement **LinkedList** class for this problem

1. Create an **Employee** class which will have details like EmployeeNo, EmployeeName and Address. You should pass value for **EmployeeNo, EmployeeName and Address** through constructor.
2. Create a method addInput( ) which will add employee details to LinkedList.
3. Create method display( ) which should display all data from LinkedList using forward and reverse order using Iterator and ListIterator interfaces.
4. Note: addInput( ) and display( ) should not be member functions of Employee class.
   1. Create a Phone Book having user interface like,
5. Add new phone book entry
6. Search Phone Number
7. Quit

Option i :it allows add name and Phone no.

Option ii: it must take name as input from the user and based on that it should return phone No.

Option iii: will terminate the program.

**Note:** Use HashMap to store phone book entries.