**OOPS LAB**

**Semester: First Year: 2018-2019**

**Practice Exercises**

1. Write a program to accept a number N and check whether it is positive, negative, or zero.
2. Write a program to accept two numbers and print the greatest among two. Extend it to 3 and more numbers.
3. Write a program to accept two numbers and display the even and odd numbers. Extend this program for a series of numbers. And check if their sum is even or odd.
4. Write a program to print all the alternate numbers between a given range of numbers.
5. Write programs to print different patterns.
6. Write a function that returns the last digit of an entered number. Extend it to print last two digits of the entered number.
7. Write a program to print the sum of last 2 digits of the entered number. Extend it to print the sum of last N digits of the entered number.
8. Write a program to count the number of repeated digits in an entered number.
9. Write a program to count the number of unique digits in an entered number.
10. Write a program to count the sum of all the even digits.
11. Write a program to count the sum of all the odd digits.
12. Write a program to give a choice of computing the sum of either the even or odd numbers.
13. Write a program to find the weight of the given hill pattern.
14. Write a program to return the second word of a given string in upper case. Extend the problem to print any word in upper case.
15. Write a program to check whether a given string is a palindrome. Extend it to check for any word in a given sentence.
16. Write a program to find the weight of a given string. Extend the program to find the weight of a given sentence.
17. Write a program to display all unique characters in the string. Extend it calculate the frequency of a character in a given string. To calculate the frequency of any given character in the String. Calculate the frequency of all unique characters in the string.
18. Given a sentence, Make groups of words of same length.
19. Given two sentences, Display the words present in both the sentences. Extend it to create groups of same words. Extend to group words with same frequencies.
20. Write a program to initialize an Array and find the max and min number in the array. Extend it to find the largest two numbers and smallest two numbers.
21. Write a program to find the common numbers in two given arrays.
22. Write a program to find the frequency of occurrence of a given number in the array. Extend it to find the frequency of each number in the array. Find the number with highest frequency and remove that number from the array.
23. Write a program to find the 2nd largest number in a 3X3 array. Extend this program to find the Nth largest/smallest number.
24. Define a class Shape, with overloaded constructors compute the area of different geometrical shapes know to you.
25. Write a program using class and object concept that prints the multiplication table for the input number.
26. Suppose that a class Employee is defined as follows:

Class Employee

{

String lastName;

String firstName;

double hourlyWage;

int yearswithCompany;

}

Suppose that data about 100 employees is already stored in an array.

Employee []employeeData = new Employee [100];

Write a code segment that will output the firstName, lastName, and hourly wage of each employee who has been with the company for 20 years or more.

1. Rational numbers are numbers that can be represented as a fraction p / q where p is an integer number and q is a positive integer (q != 0).

Design and implement a Java class Rational Number for representing such numbers. Implement methods to add und multiply rational numbers. Implement a method for return the value of a rational number as a double value. Make sure that the numerator p and denominator q do not have common divisors in your implementation. Use the algorithm for calculation the greatest common divisor to ensure this property.

Mind, that zero has a unique integral representation in your implementation.

1. Create a Box class with three attributes: length, width, height. Create three constructors with the prototype as

Box()

Box(int length, int width, int height)

Box(Box b)

Write all the get and set methods. Write the method for volume inside the class (long volume()). Create three boxes (objects) of various sizes. Compute the volumes and find the largest one.

1. Create a class called as Point. It has two integers: x and y. Write three constructors

Point() //will assign x=y=0;

Point(int x) //will assign y=0

Point(int x, int y)

Write a method that finds the distance between two points (two Point objects). Method: long distance(Point p2).

Write another class called TestPoint. Inside the main method create three Point objects. Find the distances between each one of them.

1. Write a program to extend the Vehicle class. Vehicle encapsulates information about vehicles, including the number of passengers they can carry, their fuel capacity, and fuel consumption rate. We can use the Vehicle class as a starting point from which more specialized classes are developed. For example, one type of vehicle is a truck. An important attribute of a truck is its cargo capacity. Thus, to create a Truck class, you can extend Vehicle, adding an instance variable that stores the carrying capacity. In the process, the instance variables in Vehicle will be made private, and accessor methods are provided to get and set their values. Particularly try to demonstrate different types of inheritances.
2. Write the below program.

Box.java

i) Create a java file named Box.java

ii) The class Box has three dimensions (w, l, h)

Main.java

i) Create a java file (Main.java) with class name “Main”

ii) Write main method inside class Main

iii) Create an array of Box classes (say 5 Box objects)

iv) Dimensions of each box would be (1, 1, 1), (2, 2, 2), ...

v) Display the width (w) of all the objects using a loop construct.

1. A Stack is a data structure which would hold a certain number of elements. It is a FILO (First In Last Out) data structure. It provides the following operations

a) push - void push(Box b)

b) pop – Box pop()

Stack.java

i) Create a java file named Stack.java

ii) Create an array of Box objects (say 10 Box objects)

iii) Implement push operation

iv) Implement pop operation

Main.java

i) Create a java file (Main.java) with class name “Main”

ii) Write main method inside class Main

iii) Create a Stack object

iv) Push three Box objects (1,2,3), (2,3,4) and (3,4,5) into the Stack

v) Pop two Box objects from the Stack

1. Write the below program on Inheritance.

Box.java

i) Add the method printFields (void printFields()) to the Box class. It prints the fields w, l, and h in order.

BoxWeight.java

i) Create a java file named BoxWeight.java

ii) The class BoxWeight is an extension of the class Box

iii) It has a field called wg (weight)

iv) Write a default constructor for this class.

a) Can you access/modify the variables w, l and h? If so, why?

b) Can you access/modify any of the methods of class Box?

1. Write the below program using super keyword.

Box.java

i) Write a constructor that initializes the values of w, l, and h.

Box(int w, int l, int h) {...}

BoxWeight.java

i) In the constructor for BoxWeight class, call any of the Box class constructor using super keyword.

ii) Create a couple of more constructors

Main.java

i) Create a java file (Main.java) with class name “Main”

ii) Write main method inside class Main

iii) Create an object of class BoxWeight passing the arguments w, l, h and wg.

iv) Call the method “printFields” to print the values of w, l and h.

1. Is it possible to have same “printFields” method to print wg as well.
2. Overriding

BoxWeight.java

i) Add a method printFields (void printFields()) to the BoxWeight class. It prints the fields w, l, h, wg in order.

Main.java

i) Create a java file (Main.java) with class name “Main”

ii) Write main method inside class Main

iii) Create an object of class BoxWeight passing the arguments w, l, h and wg.

iv) Call the method “printFields”.

a) What does it print?

Now, modify the printFields method of BoxWeight to add super keyword for calling base class method printFields.

1. Abstract Classes

Box.java

i) Create a method named rate (double rate()) that returns the total price of purchasing this box. Let the rate method be abstract.

BoxWeight.java

i) Implement rate method

Rate of a Box = Weight.

BoxColor.java

i) Create a java file named BoxColor.java

ii) The class BoxColor is an extension of the class Box

iii) It has a field called color. It can take the values {BLUE, RED, BLACK}

iv) Implement the rate method

If color is BLUE, then rate = 15

If color is RED, then rate = 25

If color is BLACK, then rate = 35

Main.java

i) Create a java file (Main.java) with class name “Main”

ii) Write main method inside class Main

iii) Create an object of class BoxWeight passing the arguments w, l, h and wg.

iv) What is the price of this Box object?

a) Is it possible to create an object of class Box?

1. Object class

Main.java

i) Create a java file (Main.java) with class name “Main”

ii) Write main method inside class Main

iii) Create an instance of BoxWeight class

iv) The method “String toString()” is defined in class Object. Call this method on the BoxWeight object.

a) What is the value of this string?

Box.java

i) Add the method “String toString()” to the Box class

ii) The above method should return a concatenation of w, l, and h. (e.g. “w=5; l=10; h=7”)

1. protected is an access specifier for variables, methods, and nested classes.

Do the following for Box.java

i) Move the file (Box.java) to a directory “dir1”

ii) Change the access specifier of w, l, h to “protected”

Do the following for BoxWeight.java

i) Move the file (BoxWeight.java) to a directory “dir2”

Can you access/modify the variables w, l, h?

Answer the following questions.

i) What happens if the access specifier for “w” is private?

ii) What happens if the access specifier for “w” is public?

iii) What happens if the access specifier for “w” is default?

iv) Write a protected method in Box.java. Can this method be accessed/overridden in BoxWeight.java?

1. (i) Create a package test1 and write a class in that file that contains a function to swap two numbers without using a third variable. Create another file and import the package to test the function.

(ii) In the class above, declare a function toPrint() to print the swapped numbers. Declare it as private, public or protected and observe.

1. (i) Create a package sample and declare a variable inside it. Create another file and import the package to print the variable.

(ii) In the above exercise try to print the variable when it is specified as public, private, protected and default.

(iii) Create an interface Animal which includes eat() and travel() as their abstract methods. Create another file which implements this interface.

1. What happens if all the methods are not defined here?
2. What if the class is declared abstract?
3. WAP to implement various operations on complex numbers (addition, subtraction, multiplication, division, conjugate etc.).
4. Foo Corporation needs a program to calculate how much to pay their hourly employees. The US Department of Labor requires that employees get paid time and a half for any hours over 40 that they work in a single week. For example, if an employee works 45 hours, they get 5 hours of overtime, at 1.5 times their base pay. The State of Massachusetts requires that hourly employees be paid at least $8.00 an hour. Foo Corp requires that an employee not work more than 60 hours in a week.

Rules: -

* + An employee gets paid (hours worked) × (base pay), for each hour up to 40 hours.
  + For every hour over 40, they get overtime = (base pay) × 1.5.
  + The base pay must not be less than the minimum wage ($8.00 an hour).
  + If it is, print an error. If the number of hours is greater than 60, print an error message.

Create a new class called “FooCorporation”. Write a method that takes the base pay and hours worked as parameters, and prints the total pay or an error. Write a main method that calls this method for each of these employees:

|  |  |  |
| --- | --- | --- |
|  | Base Pay | Hours Worked |
| Employee 1 | $7.50 | 35 |
| Employee 2 | $8.20 | 47 |
| Employee 3 | $10.00 | 73 |

1. Write a procedure, called isPalindrome, that takes a string as an argument and returns true if the string is a palindrome, that is, if the string is identical to the reversed string. It should return false otherwise.

Some examples are:

'able was I ere I saw elba'

Note that this works even with the spaces.

1. Define a class FruitSalad with class attributes fruits, which is initially ['melons', 'pineapples'] and servings which is initially 4.

Write a constructor that takes arguments ingredients (a list of strings) and numservings (an integer) and stores the supplied values in instance attributes fruits and servings (the servings remaining) respectively.

Write a method that returns a string containing the number of remaining servings and the fruits in the fruit salad. The string should look like this:

"2 servings of fruit salad with ['bananas', 'apples']"

Write a method add that takes a string as an argument and appends it to the end of the list fruits.

Finally, write a method serve of no arguments that returns 'enjoy' if it has been called a number of times that is less than or equal to the value of numservings supplied when the associated instance was created, or 'sorry' otherwise. It should update the servings of the instance, make sure that this variable never becomes negative.

1. Create class named **Cuboid** with fields length, width and height. Calculate volume of the cuboid.

Class Cuboid contains following functions:

set\_height( int l): sets the length of the cuboid.

set\_width(int w): sets width of the cuboid.

set\_height(int h): sets height of the cuboid.

volume (): Prints the volume of the cuboid.

Complete the methods.

1. Create class named **Cuboid** with fields length, width and height. Calculate volume of the cuboid. Class Cuboid contains following functions:

set\_length(int l): sets the length of the cuboid.

set\_width(int w): sets width of the cuboid.

set\_height(int h): sets height of the cuboid.

volume (): Prints the volume of the cuboid.

Create one object of this and initialize values. Create another class **Cub** in which there is method:

PrintData(Cuboid ob): It takes object of cuboid class and print its length, width and height.

Call PrintData method from main class.

1. Write a program in Java to remove duplicate value nodes from a sorted linked list.
2. Create a class Teacher which stores the name, designation and the courses handled by the teachers. Create a singly linked list which stores the data of several teachers and write a menu driven program which allows the user to enter the data about a certain teacher, remove the data and also search for the data. Also create a sort by method where the user passes the argument as name or courses, and you return the details sorted by the passed parameter. For e.g.- if user enters the argument as name, you return all the details sorted by the faculty names. Also create a user-defined exception class which handles the necessary exceptions.
3. You are given an interface AdvancedArithmetic which contains a method signature int divisorSum (int n). Also create an abstract class Calculator which has methods for addition, substraction, multiplication, division. You need to write a class called MyCalculator which implements the interface and extends the abstract class. divisorSum function just takes an integer as input and return the sum of all its divisors. For example, divisors of 6 are 1, 2, 3 and 6, so divisorSum should return 12. The value of n will be at most 1000. Create a user defined exception class which handles the exceptions if the value of n will be at most 1000.
4. Create user defined exception for bank to handle InSufficientFundException. For example you have one savings account in any Bank and you have 60 dollars in your account and min balanced to be maintained is 50 dollars. You attempted to withdraw 50 dollars from your account. Then your program has to display message related to insufficient funds.
5. You provide a web form for users to fill in and submit. But in case there are a lot of conditions to be handled and the conditions keeps changing periodically, you use exception handling to simplify the process. For example password condition checking, DOB checking, e-mail format checking, phone number digits checking etc.
6. Write menu driven arithmetic calculator program by including exception handling such as number format, divide by zero etc.
7. Create a base class Calculator which has methods defined for add and subtract which deals with addition, subtraction, multiplication and division of 2 numbers. Create a subclass MyCalculator which extends Calculator and perform overloading and overriding on the above methods.
8. Given a double-precision number, denoting an amount of money, use the NumberFormat class' getCurrencyInstance method to convert into the US, Indian, Chinese, and French currency formats. Then print the formatted values as follows:

US: formattedPayment

India: formattedPayment

China: formattedPayment

France: formattedPayment

where is formatted according to the appropriate Locale's currency.

Note: India does not have a built-in Locale, so you must construct one where the language is en (i.e., English).

**Input Format**

A single double-precision number denoting.

**Constraints**

0 <= payment <= 10^9.

**Output Format**

On the first line, print US: u where u is payment formatted for US currency.

On the second line, print India: i where i is payment formatted for Indian currency.

On the third line, print China: c where c is payment formatted for Chinese currency.

On the fourth line, print France: f, where f is payment formatted for French currency.

Create a user defined exception class to handle the constraint.

1. Implement a word frequency counter. Accept sentences from user in a loop (until -1) while storing them in an ArrayList. Iterate over the ArrayList to go over sentences one by one while storing the word frequency in a HashMap. Finally print the frequency of occurrence of each word.
2. In the city of Springfield, home of the ever-famous Simpson family, whenever a person votes in a city election, his/her name is added to a list of voters. This action is important because several nefarious residents of Springfield, including Mayor Quimby himself, have been known to vote more than once. To curb ballot stuffing, a person’s name is validated (the list is checked) before he/she is allowed to cast a vote. If a person has already voted, he/she is barred from voting a second time. Write an application that adds a name (String) to a list of voters and also performs rapid lookup when a potential voter arrives at the polls.
3. Write a Java program to reverse a string (Use StringTokenizer class for this exercise).
4. Each year, Sleepy Hollow Elementary School holds a “Principal for a Day” lottery. A student can participate by entering his/her name and ID number into a pool of candidates. The winner is selected randomly from all entries. Each student is allowed one entry.

Implement a class ‘StudentLottery’ with methods that

* enter students in the “Principal for a Day” lottery, and
* pick a winner randomly from the entries.

The application should check that no student enters the lottery more than once.

1. Find all possible IP address from the given input string containing only digits, IPv4 addresses are usually represented in dot-decimal notation, consisting of four decimal numbers, each ranging from 0 to 255, separated by dots, e.g., 172.16.254.1. Each part represents a group of 8 bits (octet) of the address

For example:   
Given "25525511135",

Return ["255.255.11.135", "255.255.111.35"]. (Order does not matter)

1. The Calendar class is an abstract class that provides methods for converting between a specific instant in time and a set of calendar fields such as YEAR, MONTH, DAY\_OF\_MONTH, HOUR, and so on, and for manipulating the calendar fields, such as getting the date of the next week.

You are given a date. You need to write the method, get day, which returns the *day* on that date. For example, if you are given the date August 14th 2017, the method should return Monday as the day on that date. The input should contain a single line of input containing the space separated month, day and year, respectively, in MM|DD|YYYY format.

1. Given an expression string exp, write a program to examine whether the pairs and the orders of “{“,”}”,” (“,”)”,” [“,”]” are correct in exp.

For example, the program should print true for exp = “[()]{}{[()()]()}” and false for exp = “[(])”

Hint: Use stack class.

1. Write a Java program to remove duplicate elements from an arraylist with and without using collections (with and without using Linkedhashset)
2. Sort ArrayList in Descending Order using Comparator and Collections.
3. Write a Java program to remove value from hashmap. (Hint - import Hashmap, search for remove function in Object class).
4. Write a code to reverse the string "Collection" using the Stack Class in Collection framework.
5. Create a linked list (1,9,10,4,37,2,65) using the add() method. Print the linked list after deleting the middle number using the remove() method. This has to be done using the LinkedList class in collection framework.
6. Find the maximum element in the sliding window, given input array of n elements you have to find maximum element in an window of size k example:

Window position Max

--------------- -----

[1 3 -1] -3 5 3 6 7 3

1 [3 -1 -3] 5 3 6 7 3

1 3 [-1 -3 5] 3 6 7 5

1 3 -1 [-3 5 3] 6 7 5

1 3 -1 -3 [5 3 6] 7 6

1 3 -1 -3 5 [3 6 7] 7

Print the maximum element in the each sliding window position. i.e. 3 3 5 5 6 7

Note: You may assume k is always valid, i.e.: 1 ≤ k ≤ input array's size for non-empty array. Hint: use dequeue class.