Middlesex University London

2021

Workbook

Student Name:…………………………………………………………………

Student ID:……………………………………………………………………

Tutor Name:……………………………………………………………………

Department of Computer Science

Exercises organisation

These exercises are designed to consolidate concepts covered in scheduled workshops to give you a practical insight into programming for data communication.

This workbook contains four main sections; each section runs for set number of weeks as listed below and contains number of coding activities:

- Section 1: Introduction to Programming, runs for 7 lectures.
- Section 2: Object-Oriented Thinking, runs for 4 lectures.
- Section 3: User Interface and Error Handling, runs for 4 lectures.
- Section 4: Input, Process, Storage, Output, runs for 2 lectures.

In this module, you will learn key programming concepts to build a solid foundation for solving communication problems using JAVA programming language.

Contents

SECTION 1

Fundamentals of Programming

Lecture 1 Introduction

Threshold SOB 1: NetBeans create, export and import projects.
Show the ability to create, export and import project in NetBeans

Threshold SOB 2: Single line and multiline comments
Show the ability to comment out code using single and block (multiline) comments.

Threshold SOB 3: Printing to the terminal

Write a program that displays “Welcome to Java”, “Learning Java Now” and “Programming is fun”

Support Material:

Chapter 1, page 36 Printing to console

Typical SOB 4: Printing to the terminal

Write a program that displays your personal credentials such as: name, surname, dob, age, the course you choose to study on.

Consider using the System.out.print , the System.out.println method and as many as you know of the escape sequences.

Support Material:

Chapter 1, page 36 Printing to console

Chapter 4, page 148 (Table 4.5) Escape Sequences

Lecture 2 Input-Output

Threshold SOB 5: char variable concatenation

Create a program that takes 5 char variables using Scanner then concatenate them into one word and print it out to screen.

Support Material:

Chapter 4, page 154 String concatenation

Threshold SOB 6: String variable concatenation

Create a program that takes 3 String variables using Scanner then concatenate them into one sentence and print it out to screen.

Support Material:

Chapter 2, page 58. Chapter 4, page 154: String concatenation

Lecture 3 Primitive data types

Threshold SOB 7: Value conversion using double

Write a program that reads miles in a double value from the console, then converts it to kilometres and displays the result. The formula for the conversion is as follows 1 mile = 1.6 kilometres

Support Material:

Chapter 2, page 59 Reading input from console

Typical SOB 8: Circle area

Show ability to use double variables by creating two double variables where one will store value of PI (3.14159), and the other will store circle radius of 32.32.

Using formula A = PI * R^2 calculate an area of the circle and print it to console.

Support Material:

Chapter 2, page 58, Chapter 4, page 154: String concatenation

Excellent SOB 9: Math class and PI constant

Show ability to use Math class to calculate area of circle, by using PI constant value, and exponent function of the class.

Support Material:

Chapter 2, page 59 Reading input from console

Chapter 4.2, page 142 Math Class and usage

Lecture 4 Selections

Threshold SOB 10: Coin toss

Write a program that lets the user guess whether the flip of a coin results in heads or tails. The program randomly generates an integer 0 or 1, which represents head or tail. The program prompts the user to enter a guess and reports whether the guess is correct or incorrect.

Support Material:

Chapter 2, page 55 Reading input from console

Chapter 3, page 100 If statement

Chapter 9, page 359 Random class

Threshold SOB 11: Number of digits in a number

Write a program which display the number of digits in a given number.

The program should use Scanner to ask for a number.

If the Input number is for example: 2001

The expected Output would be:

"This number has 4 digits"

Support Material:

Chapter 2.3, page 55 Reading input from console

Chapter 3, page 102 If else statement

Typical SOB 12: Shipping company

A shipping company uses the following function to calculate the cost (in dollars) of shipping based on the weight of the package (in pounds). It cost 2.5, if weight is between 0 and 2 It cost 4.5, if weight is between 2 and 4 It cost 7.5, if weight is between 4 and 10 It cost 10.5, if weight is between 10 and 20 Write a program that prompts the user to enter the weight of the package and display the shipping cost. If the weight is greater than 20, display a message "the package cannot be shipped."

Support Material:

Chapter 2, page 59 Reading input from console

Chapter 3, page 100 If statement, if-else statement

Excellent SOB 13: What month is this?

Write a program that randomly generates an integer between 1 and 12 and displays the English month name January, February, …, December for the number 1, 2, …, 12, accordingly.

Support Material:

Chapter 2, page 59 Reading input from console

Chapter 3, page 100 If statement

Chapter 9, page 359 Random Class

Lecture 5 Loops

Threshold SOB 14: To fail or not to fail.

Write a program that prompts a student to enter a Java score. If the score is greater or equal to 60 then display "you pass the exam", otherwise display "you don't pass the exam". You program ends with input -1.

Sample run:

```
Enter your score: 80 (Enter)
You pass the exam

Enter your score: 59
You did not pass the exam

Enter your score: -1
No numbers are entered expect 0.
```

Support Material:

Chapter 2, page 55 Reading input from console

Chapter 3, page 102 If statement

Chapter 5, page 189 For loop

Threshold SOB 15: Student group

Write a program that prompts the user to enter the number of students and each student's name and score, and finally displays the student with the lowest score and the student with the second-lowest score.

Support Material:

Chapter 2, page 59 Reading input from console

Chapter 3, page 100 If statement

Chapter 5, page 195 For loop

Typical SOB 16: Celsius vs. Fahrenheit

Write a program that displays the following table (note that Fahrenheit = Celsius * 9/5 + 3.2):

Celsius	Fahrenheit
0	32.0
2	35.6
…	
98	208.4
100	212.0

Support Material:

Chapter 2, page 59 Reading input from console

Chapter 3, page 100 If statement

Chapter 5, page 195 For loop

Excellent SOB 17: Student loan…

Suppose that the tuition for a university is $10,000 this year and increases 6% every year. After a year, the tuition will be $10,600. Write a program that computes the tuition in ten years and the total cost of four years' worth of tuition after the tenth year.

Support Material:

Chapter 2, page 59 Reading input from console

Chapter 5, page 195 For loop

Lecture 7 Methods

Threshold SOB 18: I will walk 500 miles

Create two methods to perform following actions:

- Convert from Mile to Kilometre
  - public static double mileToKilometer(double mile)
- Convert from Kilometre to Mile
  - public static double kilometerToMile(double kilometre)

The formula for the conversion is:

1 mile = 1.6 kilometres

Write a test program that invokes these methods to display the following tables:

Miles	Kilometres	Kilometres	Miles
1	1.609	20	12.430
2	3.218	25	15.538
9	14.481	60	37.290
10	16.090	65	40.389

Support Material:

Chapter 5, page 195 For loop

Chapter 6, page 228 Method declaration

Typical SOB 19: Count my ABC

Write a method that counts the number of letters in a string using the following header:

public static int countLetters(String s)

Write a test program that prompts the user to enter a string and displays the number of letters in the string.

Support Material:

Chapter 4, page 148 String length

Chapter 6, page 222 Method declaration

Excellent SOB 20: Password security

Some websites impose certain rules for passwords. Write a method that checks whether a string is a valid password. Suppose the password rules are as follows:

- A password must have at least ten characters.
- A password consists of only letters and digits.
- A password must contain at least three digits.

Write a program that prompts the user to enter a password and displays Valid Password if the rules are followed or Invalid Password otherwise.

Support Material:

Chapter 3, page 94 If statement

Chapter 4, page 148 String length

Chapter 6, page 222 Method declaration

SECTION 2

Object-Oriented Thinking

Lecture 8 Array

Threshold SOB 21: Vector method summation

Write two methods that return the sum of an array with the following headers:

public static int sum(int[] array)

public static double sum(double[] array)

Write a test program that prompts the user to enter ten double values, invokes this method, and displays the sum value.

Support Material:

Chapter 2 page 59, Reading input from console

Chapter 4 page 142, Math Class and usage

Chapter 5 page 189, For loop

Chapter 6.1 page 222, Method declaration

Typical SOB 22: Largest integer of array

Write a method that returns the index of the largest element in an array of integers.

If the number of such elements is greater than 1, return the largest index.

Use the following header:

public static int indexOfLargestElement(double[] array)

Write a test program that prompts the user to enter ten numbers, invokes this method to return the index of the largest element, and displays the index.

Support Material:

Chapter 2.3 page 59, Reading input from console

Chapter 3 page 97 If statement

Chapter 4.2 page 142, Math Class and usage

Chapter 5.4 page 194, For loop

Chapter 6.1 page 228, Method declaration

Excellent SOB 23: Clean row

Write a method that returns a new array by eliminating the duplicate values in the array using the following method header:

public static int[] eliminateDuplicates(int[] list)

Write a test program that reads in ten integers, invokes the method, and displays the result.

Support Material:

Chapter 2.3 page 59, Reading input from console

Chapter 3 page 100, If statement

Chapter 4.2 page 142, Math Class and usage

Chapter 5.4 page 195, For loop

Chapter 6.1 page 228, Method declaration

Lecture 9 2D arrays

Threshold SOB 24: Matrix method summation

Write a method that returns the sum of all the elements in a specified row in a matrix using the following header:

public static double sumRow(double[][] m, int rowIndex)

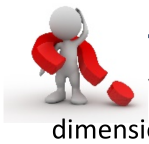
Write a test program that reads a 3-by-4 matrix and displays the sum of each row.

Support Material:

Chapter 3 page 100, If statement

Chapter 5.4 page 195, For loop

Chapter 6.1 page 228, Method declaration



Typical SOB 25: Matrix the smallest value

Write the following method that returns the location of the smallest element in a two-dimensional array.

public static int[] locateSmallest(double[][] a)

The return value is a one-dimensional array that contains two elements. These two elements indicate the row and column indices of the smallest element in the two-dimensional array. Write a test program that prompts the user to enter a two-

dimensional array and displays the location of the smallest element in the array.

Support Material:

Chapter 3 page 100, If statement

Chapter 5.4 page 195, For loop

Chapter 6.1 page 228, Method declaration

Excellent SOB 26: Binary counting

Write a program that randomly fills in 0s and 1s into a 5-by-5 matrix, prints the matrix, and finds the first row and column with the most 1s.

Here is a sample run of the program:

```
01101
01011
10010
11111
00101


The largest row index: 3

The largest column index: 4
```

Support Material:

Chapter 3 page 100, If statement

Chapter 5.4 page 195, For loop

Chapter 6.1 page 228, Method declaration

Chapter 9.6 page 359, Random class

Lecture 10 Classes

Threshold SOB 27: Account class

Design a class named Account that contains:

- A private int data field named id for the account (default 0).
- A private double data field named balance for the account (default 0).
- A private double data field named annualInterestRate that stores the current interest rate (default 0). Assume all accounts have the same interest rate.
- A private Date data field named dateCreated that stores the date when the account was created.
- A no-arg constructor that creates a default account.
- A constructor that creates an account with the specified id and initial balance.
- The accessor and mutator methods for id, balance, and annualInterestRate.
- The accessor method for dateCreated.
- A method named getMonthlyInterestRate() that returns the monthly interest rate.
- A method named getMonthlyInterest() that returns the monthly interest.
- A method named withdraw that withdraws a specified amount from the account.
- A method named deposit that deposits a specified amount to the account.

Write a test program that creates an Account object with an account ID of 1122, a balance of $20,000, and an annual interest rate of 4.5%. Use the withdraw method to withdraw $2,500, use the deposit method to deposit $3,000, and print the balance, the monthly interest, and the date when this account was created.

Support Material:

Chapter 6.1 page 228, Method declaration

Chapter 9.2 page 346, Class declaration

Typical SOB 28: Fan class

Design a class named Fan to represent a fan.

The class contains:

- Three constants named SLOW, MEDIUM, and FAST with the values 1, 2, and 3 to denote the fan speed.
- A private int data field named speed that specifies the speed of the fan (the default is SLOW).
- A private boolean data field named on that specifies whether the fan is on (the default is false).
- A private double data field named radius that specifies the radius of the fan (the default is 5).
- A string data field named color that specifies the color of the fan (the default is blue).
- The accessor and mutator methods for all four data fields.
- A no-arg constructor that creates a default fan.
- A method named toString() that returns a string description for the fan. If the fan is on, the method returns the fan speed, color, and radius in one combined string. If the fan is not on, the method returns the fan color and radius along with the string “fan is off” in one combined string.

Write a test program that creates two Fan objects. Assign maximum speed, radius 12, color green, and turn it on to the first object. Assign medium speed, radius 6, color red, and turn it off to the second object. Display the objects by invoking their toString method.

Support Material:

Chapter 6.1 page 228, Method declaration

Chapter 9.2 page 346, Class declaration

Excellent SOB 29: Calendar class

Java API has the GregorianCalendar class in the java.util package, which you can use to obtain the year, month, and day of a date. The no-arg constructor constructs an instance for the current date, and the methods get(GregorianCalendar.YEAR), get(GregorianCalendar.MONTH), and get(GregorianCalendar.DAY_OF_MONTH) return the year, month, and day.

- Write a program to perform two tasks:
- Display the current year, month, and day.

The GregorianCalendar class has the setTimeInMillis(long), which can be used to set a specified elapsed time since January 1, 1970. Set the value to 1234567898765L and display the year, month, and day.

Support Material:

Chapter 6.1 page 228, Method declaration

Chapter 9.2 page 346, Class declaration

Chapter 13.4 page 529, Calendar class

Lecture 11 Object-Oriented Thinking

Threshold SOB 30: Personal integer class

Design a class named MyInteger.

The class contains:

- An int data field named value that stores the int value represented by this object.
- A constructor that creates a MyInteger object for the specified int value.
- A getter method that returns the int value.
- The methods isEven(), isOdd(), and isPrime() that return true if the value in this object is even, odd, or prime, respectively.
- The static methods isEven(int), isOdd(int), and isPrime(int) that return true if the specified value is even, odd, or prime, respectively.
- The static methods isEven(MyInteger), sOdd(MyInteger), and isPrime(MyInteger) that return true if the specified value is even, odd, or prime, respectively.
- The methods equals(int) and equals(MyInteger) that return true if the value in this object is equal to the specified value.
- A static method parseInt(char[]) that converts an array of numeric characters to an int value.
- A static method parseInt(String) that converts a string into an int value.

Write a client program that tests all methods in the class.

Support Material:

Chapter 4.2 page 142, Math Class and usage

Chapter 6.1 page 228, Method declaration

Chapter 9.2 page 346, Class declaration

Typical SOB 31: Personal Date class

Design a class named MyDate.

The class contains:

- The data fields year, month, and day that represent a date. month is 0-based, i.e., 0 is for January.
- A no-arg constructor that creates a MyDate object for the current date.
- A constructor that constructs a MyDate object with a specified elapsed time since midnight, January 1, 1970, in milliseconds.
- A constructor that constructs a MyDate object with the specified year, month, and day.
- Three getter methods for the data fields year, month, and day, respectively.
- A method named setDate(long elapsedTime) that sets a new date for the object using the elapsed time.

Write a test program that creates two MyDate objects (using new MyDate() and new MyDate(34355555133101L)) and displays their year, month, and day.

(Hint: The first two constructors will extract the year, month, and day from the elapsed time. For example, if the elapsed time is 561555550000 milliseconds, the year is 1987, the month is 9, and the day is 18. You may use the

Support Material:

Chapter 4.2 page 142, Math Class and usage

Chapter 6.1 page 228, Method declaration

Chapter 9.2 page 346, Class declaration

Chapter 13.4 page 529, Calendar class

Excellent SOB 32: Personal Date class

The String class is provided in the Java library.

Provide your own implementation for the following methods (name the new class MyString1):

public MyString1(char[] chars);

public char charAt(int index);

public int length();

public MyString1 substring(int begin, int end);

public MyString1 toLowerCase();

public boolean equals(MyString1 s);

public static MyString1 valueOf(int i);

Support Material:

Chapter 3 page 100, If statement

Chapter 5.4 page 195, For loop

Math Class and usage

Chapter 6.1 page 228, Method declaration

Chapter 9.2 page 346, Class declaration

Chapter 10.10 page 410, String class

SECTION 3

User Interface and Error Handling

Lecture 12 Inheritance, Polymorphism, ArrayList

Threshold SOB 33: ArrayList number sorting

Write the following method that sorts an ArrayList of numbers:

public static void sort(ArrayList<Integer> list)

Write a test program that prompts the user to enter 5 numbers, stores them in an array list, and displays them in decreasing order.

Design a class named Person and its two subclasses named Student and Employee.

Make Faculty and Staff subclasses of Employee. A person has a name, address, phone number, and email address. A student has a class status (freshman, sophomore, junior, or senior). Define the status as a constant. An employee has an office, salary, and date hired. Use the MyDate class defined in Programming.

Support Material:

Chapter 1 page 32, Printing to console

Chapter 2.3 page 59, Reading input from console

Chapter 3 page 100, If statement

Chapter 11 page 456, ArrayList class

Threshold SOB 34: Person of Interest

Design a class named Person and its two subclasses named Student and Employee.

Make Faculty and Staff subclasses of Employee. A person has a name, address, phone number, and email address. A student has a class status (freshman, sophomore, junior, or senior). Define the status as a constant. An employee has an office, salary, and date hired. Use the MyDate class defined in Programming.

Support Material:

Chapter 1 page 32, Printing to console

Chapter 2.3 page 59, Reading input from console

Chapter 3 page 100, If statement

Chapter 11 page 456, ArrayList class

Typical SOB 35: ArrayList size

Write the following method that returns the minimum value in an ArrayList of integers. The method returns null if the list is null or the list size is 0.

public static Integer min(ArrayList<Integer> list)

Write a test program that prompts the user to enter a sequence of numbers ending with 0, and invokes this method to return the smallest number in the input.

Support Material:

Chapter 1 page 32, Printing to console

Chapter 2.3 page 59, Reading input from console

Chapter 3 page 100, If statement

Chapter 11 page 456, ArrayList class

Excellent SOB 36: Geometric object

Design a class named Triangle that extends GeometricObject.

The class contains:

- Three double data fields named side1, side2, and side3 with default values 1.0 to denote three sides of the triangle.
- A no-arg constructor that creates a default triangle.
- A constructor that creates a triangle with the specified side1, side2, and side3.
- The accessor methods for all three data fields.
- A method named getArea() that returns the area of this triangle.
- A method named getPerimeter() that returns the perimeter of this triangle.
- A method named toString() that returns a string description for the triangle.

The toString() method is implemented as follows:

```
return "Triangle: side1 = " + side1 + " side2 = " + side2 +  " side3 = " + side3;
```

Write a test program that prompts the user to enter three sides of the triangle, a color, and a Boolean value to indicate whether the triangle is filled. The program should create a Triangle object with these sides and set the color and filled properties using the input. The program should display the area, perimeter, color, and true or false to indicate whether it is filled or not.

Support Material:

Chapter 6.1 page 228, Method declaration

Chapter 9.2 page 346, Class declaration

Chapter 11.1 page 434, Inheritance and polymorphism

Lecture 13 Exception handling

Threshold SOB 37: Incorrect numerical value

Write a program that prompts the user to read two integers and displays their product. Your program should prompt the user to read the number again if the input is incorrect.

Support Material:

Chapter 1 page 32, Printing to console

Chapter 2.3 page 59, Reading input from console

Chapter 12 page 476, Input exception

Typical SOB 38: Array out of bounds

Write a program that meets the following requirements:

- Creates an array with 120 randomly chosen integers.
- Prompts the user to enter the index of the array, and then displays the corresponding element value. If the specified index is out of bounds, display the message Out of Bounds.

Support Material:

Chapter 1 page 32, Printing to console

Chapter 2.3 page 59, Reading input from console

Chapter 9.6 page 353, Random class

Chapter 12.3 page 481, IndexOutOfBoundsException

Excellent SOB 39: Binary exception

Write the bin2Dec(String binary String) method to convert a binary string into a decimal number. Implement the bin2Dec method to throw a NumberFormatException if the string is not a binary string.

Support Material:

Chapter 1 page 32, Printing to console

Chapter 2.3 page 59, Reading input from console

Chapter 12.8 page 497, NumberFormatException

Lecture 14-15 Introduction to JavaFX

Threshold SOB 40: JavaFX Celsius and Fahrenheit conversion

Write a program that converts between Celsius and Fahrenheit. If you enter a value in the Celsius text field and press the Enter key, the corresponding Fahrenheit measurement is displayed in the Fahrenheit text field. Likewise, if you enter a value in the Fahrenheit text field and press the Enter key, the corresponding Celsius measurement is displayed in the Celsius text field.

Support Material:

Chapter 3 page 100, If statement

Chapter 15.3 page 619, Event source and event handling

Typical SOB 41: JavaFX numerical conversion

Write a program that converts between decimal, hex, and binary numbers. When you enter a decimal value in the decimal value text field and press the Enter key, its corresponding hex and binary numbers are displayed in the other two text fields. Likewise, you can enter values in the other fields and convert them accordingly. (Hint: Use the Integer.parseInt(s, radix) method to parse a string to a decimal and use Integer.toHexString(decimal) and Integer.toBinaryString(decimal) to obtain a hex number or a binary number from a decimal.)

Support Material:

Chapter 3 page 100, If statement

Chapter 15.3 page 619, Event source and event handling

Excellent SOB 42: Traffic lights

Write a program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio but tons selected, the light is turned on. Only one light can be on at a time. No light is on when the program starts.

Support Material:

Chapter 3 page 100, If statement

Chapter 15.3 page 619, Event source and event handling

SECTION 4

Input, Process, Storage, Output

Lecture 16 File writing, reading/ Binary input, output

Threshold SOB 43: Random file numbers

Write a program to create a file named numbers.txt if it does not exist. Write 120 integers created randomly into the file using text I/O. Integers are separated by spaces in the file. Read the data back from the file and display the data in increasing order.

Support Material:

Chapter 1 page 32, Printing to console

Chapter 3 page 100, If statement

Chapter 5.4 page 189, For loop

Chapter 9.6 page 359, Random class

Chapter 12.11 page 502, File input and Output

Typical SOB 44: Word count

Write a program that will count the number of characters, words, and lines in a file. Words are separated by whitespace characters. The file name should be passed as a command-line argument.

Support Material:

Chapter 1 page 32, Printing to console

Chapter 3 page 100, If statement

Chapter 5.4 page 195, For loop

Chapter 12.11 page 502, File input and Output

Excellent SOB 45: Student in a file

Create a Student object containing student's name, surname, and student number. Write program to create student object and save that object into a file.

Support Material:

Chapter 17.6 page 728, Object I/O

Lecture 17 Networking

Threshold SOB 46: Bank loan

Write a server for a client. The client sends loan information (annual interest rate, number of years, and loan amount) to the server. The server computes monthly payment and total payment, and sends them back to the client.

Typical SOB 47: Bank loan 2.0

Revise threshold SOB. Make the client send a loan object that contains annual interest rate, number of years, and loan amount and for the server to send the monthly payment and total payment.

Excellent SOB 48: Chat application

Write a program that enables two users to chat. Implement one user as the server and the other as the client. The server has two text areas: one for entering text and the other (noneditable) for displaying text received from the client. When the user presses the Enter key, the current line is sent to the client. The client has two text areas: one (noneditable) for displaying text from the server and the other for entering text. When the user presses the Enter key, the current line is sent to the server.