

**CARDIFF UNIVERSITY
EXAMINATION PAPER**

Academic Year: 2015/2016
Examination Period: Spring
Examination Paper Number: CMT205
Examination Paper Title: Object Oriented Development with Java
Duration: 2 hours

Do not turn this page over until instructed to do so by the Senior Invigilator.

Structure of Examination Paper:

There are 6 pages.

There are 4 questions in total.

There are no appendices.

The maximum mark for the examination paper is 60 and the mark obtainable for a question or part of a question is shown in brackets alongside the question.

Students to be provided with:

The following items of stationery are to be provided:
ONE answer book.

Instructions to Students:

Answer THREE questions.

Students are permitted to introduce to the examination any textbook, any printed / handwritten notes, and other similar materials. Use of annotations, highlighting and bookmarks is permitted.

The use of calculators is permitted in this examination.

Important note: if you answer more than the number of questions instructed, then answers will be marked in the order they appear only until the above instruction is met. Extra answers will be ignored. Clearly cancel any answers not intended for marking. Write clearly on the front of the answer book the numbers of the answers to be marked.

The use of translation dictionaries between English or Welsh and a foreign language bearing an appropriate departmental stamp is permitted in this examination.

- Q1. (a) What are the values of the variables `res1`, `res2`, `res3` and `res4` after the following Java statements have been executed? [4]

```
int i = 23;
double d = 3.2;
double res1 = Math.round((double)i / 5) + d;
double res2 = (double)(d * (1/3) + i);
StringBuffer buffer = new StringBuffer("Hello" + i);
buffer.replace(2, 4, "abc");
String res3 = buffer.toString();
buffer.append("#");
buffer.reverse();
String res4 = buffer.toString();
```

- (b) Write Java code fragments for the following tasks:

- i. Given a `String str`, create a new `String` with variable name `numOnly` that keeps all the digits ('0', '1', ... '9') only and in the order they appear in `str`. [3]
- ii. Create an integer array `nums` that contains 10 random numbers in the range of 1000 to 9999 (both inclusive). [3]

- (c) Assume you are developing a Java program that manages electronic records for book borrowing from the University library. The program involves a class named `User` that represents a generic user account, a class named `Student` that represents a student user, and a class named `Staff` that represents a member of staff user.

- i. When you define these classes, what are the relationships between them? [1]
- ii. Define a method `getEntitlement` to provide a consistent way of finding out the borrowing entitlement for an individual user, assuming the entitlement for `Student` users are 12 and `Staff` users are 20. Thus for a general `User` no well defined answer can be given. Briefly explain how you would define this method and how you would implement it in these classes. You should follow object-oriented development principles and justify your choices. [3]

- (d) Assume you are developing a Java program that manages bank accounts for users. It involves a class `BankAccount` which currently involves methods `setName` to set the account name, `setDOB` to set the date of birth of the user, and `deposit` and `withdraw` methods to deposit to and withdraw from the account.

- i. To help monitor account activities, before executing any method of the `BankAccount` class starting with `set` (such as `setName`) as well as the `deposit` and `withdraw` methods, the monitoring code should be executed. To provide such monitoring code in a centralised place, what technique is useful? Briefly explain major benefits of this technique. [2]
- ii. Write simple code snippets to demonstrate how the technique can be used to allow the monitoring code in the centralised place to be run as desired. You don't need to provide the actual implementation for the monitoring code. [4]

Q2. (a) Express the following JavaFX Alert responses in Java 8 *lambda expression* form:

```
i. Alert alert = new Alert(AlertType.INFORMATION);
    alert.setTitle("Information Dialog");
    alert.setHeaderText("This is an Information Dialog");
    alert.setContentText("This is an Information Dialog");

    Optional<ButtonType> result = alert.showAndWait();

    //Traditional way to get the response value.
    // **** REPLACE THIS CODE
    if ((result.isPresent()) && (result.get() == ButtonType.OK)) {

        ButtonType OKButton = result.get();

        System.out.println("OK Button Pressed");
        System.out.println("\tButton Data = " + OKButton.getButtonData());
        System.out.println("\tButton Text = " + OKButton.getText());
    }
```

[3]

```
ii. Alert alert = new Alert(AlertType.CONFIRMATION);
    alert.setTitle("Confirmation Dialog");
    alert.setHeaderText("This is a Confirmation Dialog");
    alert.setContentText("Are you ok with this?");

    Optional<ButtonType> result = alert.showAndWait();

    // Traditional way to get the response value.
    // **** REPLACE THIS CODE
    if (result.get() == ButtonType.OK) {
        // ... user chose OK
        System.out.println("Ok Button Pressed");
    } else {
        //... user chose CANCEL or closed the dialog
        System.out.println("Dialog Cancelled");
    }
```

[5]

(b) Give a suitable fragment of Java code to illustrate how you would create a simple `RadioButton` interface where you can choose one from the following list of items: Selection 1, Selection 2, Selection 3.

You should provide appropriate *mnemonic* key selection of each item. [4]

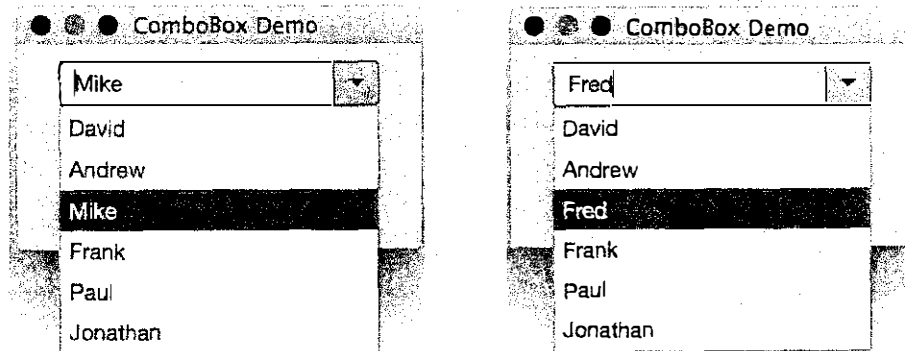
Note: Your solution to need only address creating the interface and *not* any event handling.

(Question continues on next page)

- (c) Describe using suitable fragments of JavaFX code how you would implement the following interface:

Create `ComboBox` that displays a list of forenames. When selected the forename in the list can be amended by typing in new text. The new input name from the `ComboBox` will then update the selected item in the list accordingly.

For example:



[8]

Q3. (a) Suppose we wish to store student records in a structured binary file. A record includes the following fields (name and type):

- studentID: int (the ID of the student)
- name: String (the name of the student, up to 30 characters)
- address: String (the address of the student, up to 100 characters)
- avgMark: double (the average mark for the student in the current academic year)

We further assume that studentID is allocated sequentially when they first enrolled, starting from 10,000, and the maximum studentID currently allocated is 12,000.

- i. Assuming we need to dynamically update some student records with the remaining records unchanged, which type is more suitable, sequential or random access file? Justify your answer. [2]
- ii. Work out the space (number of bytes) each record takes for a random access file. [3]
- iii. To avoid wasting space unnecessarily, how do you decide where to store each record? Write the line of code that is responsible for locating where to read/write a record with a given studentID, assuming file is the data file object. [2]

(b) Write a Java program LineCounter that finds all the text files in the current directory (referred to as ".", not including any subdirectories) with ".txt" extension. For each file found, print the filename, followed by the number of lines in the text file. For example, assuming the current directory contains abc.txt with 20 lines of text, and def.txt with 50 lines of text, the program should print:

```
abc.txt 20
def.txt 50
```

You can assume all the necessary classes from the Java standard library are imported. The following skeleton code is provided and only the code to complete the program needs to be provided. [13]

```
import java.io.*;
public class LineCounter
{
    public static void main(String[] args)
    {
        // TODO: Complete the code
    }
}
```

- Q4. (a) Write a Java program `AddServer` which implements a TCP server that sends a question of adding two random single digit numbers to the client, waits to receive a line from the client containing the number, and the server then sends back a message indicating if the answer is correct. Your program should listen to incoming **TCP** connections at the port 6000. For simplicity only one connection needs to be dealt with at a time. Every time a new connection is made, the program sends a line to the client in the format of `a+b` where `a` and `b` are single digit numbers (0, 1, 2, ... 9). The server then waits for the client to send back a line containing a single integer number. If this number gives the correct answer for `a+b`, the server sends back "Correct", otherwise it sends back "Incorrect". Once this is completed, the server will close the TCP connection.

You may assume that the content received from the client is always in the correct format but communication exceptions need to be handled properly. You can assume all the necessary classes from the Java standard library are imported.

Example:

Send: 3+8

Receive: 11

Send back: Correct

Another example:

Send: 0+3

Receive: 2

Send back: Incorrect

The following skeleton program is provided and only the code to complete the program needs to be provided. [15]

```
import java.net.*;
import java.io.*;
import java.util.*;
public class AddServer
{
    public static void main()
    {
        // Create a Server Socket object
        ServerSocket sSock = null;
        try
        {
            sSock = new ServerSocket( 6000 );
        }
        catch( IOException e )
        {
            System.err.println(e);
            System.exit(1);
        }
        // TODO: Complete the program
    }
}
```

- (b) Using code snippets if necessary, describe all the changes to the code above to use multi-threading so that the server can handle multiple connections at the same time. You can refer to the code you have given in the previous answer and don't need to repeat it. [5]