U08026 Further Object-Oriented Programming 2013–2014

Coursework 3

Value

25%

Mode

Individual work

Deadline

Sunday 15 December 2013 at 17:00 (end of Week 12).

Upload to *Turnitin* via *Moodle*.

Learning outcomes

This coursework is designed to test your attainment of the following learning outcomes:

- Evaluate the correctness and usefulness of a software system
- Analyse a complex problem, structure it, collect relevant information, consider options and recommend a course of action.
- Identify and utilise trustworthy information sources which provide programming-language specific reference material.

Introduction

The purpose of this assessment is to examine a deeper knowledge of the concepts that have been covered in classes, weekly exercises and courseworks. We are trying to prepare you for the sort of question that will come up in examinations of later modules, where you are asked to 'explain' concepts. Such questions typically give you little guidance as to what to include in your answer – unlike your courseworks, which are structured for you. For that reason the questions here do not give hints as to what is looked for in the answer – that is for you to determine. It is important for you to gain practice in answering such questions; the feedback we give you on this assessment should prove useful for your later modules.

The questions are of the form of examinations questions, but you can take your time in preparing your answers. Although you may use whatever sources you wish for preparation it is important that the answers you submit should be in your own words, as you would write in an examination – no 'copy and paste' or answers written in collusion with others. To remind you of this and to monitor it we are requiring submission via *Turnitin*.

We do not intend this assessment to take you any longer than it would have taken you to revise for an examination and then take it – say half a day. You should not write any more than would be possible in the duration of a normal examination.

As a guide, we expect that you should write 1,000 – 1,500 words.

You may write answers in note form and may use examples and diagrams where you see fit. You may draw any diagrams by hand and scan to include in the document you submit.

Rubric

- Answer *all five* questions.
- As a guide, we expect that you should write 1,000 – 1,500 words.
- Each question will be marked out of 10 and the total divided by two to give a mark out of 25.
- Answers must be in your own words, as if in an examination.

Questions – answer them all

1) Explain the principles of operation of the sorting algorithm known as *Quicksort*. Explain why it is so much quicker than a 'naïve' sorting algorithm, such as *bubble sort* or *insertion sort*.

10 marks

2) Explain the differences between the collection class *HashMap* and *TreeMap*. Indicate in what circumstances you would use each.

10 marks

3) Explain the differences between an *exception* and an *assertion*. Indicate the circumstances in which you would make appropriate use of each.

10 marks

4) Explain the advantages and disadvantages of single/multiple inheritance. Java has only single class inheritance but allows multiple implementation of interfaces. Explain, with the aid of an example, what this means and compare it with multiple inheritance.

10 marks

5) In the context of concurrency, what are meant by the terms *race condition* and *deadlock*? What features does Java offer to help with these problems?

10 marks

= 50 marks
to be divided by 2

What you have to submit to *Turnitin* (*Word*)

A *Word* (.doc or .docx) document containing your answers