Ian Kenny

September 19, 2017

Welcome to Software Workshop

The aims of the module are

- To present the fundamental concepts of imperative and object-oriented programming.
- For you to develop the skills needed to design, develop and document programs.
- For you to gain working knowledge of the Java programming language.

The overall aim is for all students to have an understanding of some key programming ideas by the end of term 1. Some students will enter the module having done lots of programming and may find that they know some or much of the material. Some students may have done no programming before at all. The aim of this module, therefore, is to get everyone to *at least* the same specific point.

Staff

Semester 1: Ian Kenny, room 232, i.s.kenny@cs.bham.ac.uk

Semester 2: Martin Escardo, room 212, m.escardo@cs.bham.ac.uk

Teaching assistants: to give you feedback on your work.

Communication

The best way to contact one of the lecturers is to email them. You can also look out for their office hours which should be shown on their office door.

For this module you can also use the Facebook group **but** do not rely on that to ask lecturers (or teaching assistants) urgent questions. The Facebook group is largely for you to talk amongst yourselves about the work on the module, but tutors will be dropping in too.

The Facebook group is linked to on Canvas. It is called '2017-2018_Software_Workshop'.

Module structure

Three lectures per week

- Monday, 1300, Aston Webb Main Lecture Theatre
- Thursday, 1500, Muirhead Tower, G15
- Friday, 1300, Aston Webb Main Lecture Theatre

Labs are spread across the week. Refer to your timetable.

THERE ARE NO LABS ON THIS MODULE IN WEEK 1 (NEXT WEEK)!

Assessment

The module has 100% continuous assessment (no exams).

The module is divided into two halves: term 1 and term 2. The part of the module we are discussing now is term 1.

Term 1 is worth 20% of the marks, term 2 is worth 80% of the marks.

Assessment is through regular programming exercises that you must submit and pass. Some of the weekly exercises will be *formative* (i.e. non-assessed). Others are *summative* (assessed). You will receive feedback on both types of assessment.

There will also be regular tests (called 'quizzes' on Canvas) that you must pass but do not count towards your mark for the module.

I will give you much more detail on assessment next week.

Content

The indicative content of the module is

- Fundamentals of imperative programming, Java introduction.
- Data types, variables, operators, selection, iteration, arrays.
- Object-oriented programming in Java: classes, methods, constructors, using objects, access modifiers.
- Inheritance and composition, inner classes.
- Polymorphism and method overriding.
- Abstract classes.
- Interface classes.
- Java collections, generics.
- Exceptions.
- File handling.
- GUI building.
- Design patterns.
- Object-Oriented software principles.

This is not an exhaustive list. A lot of other topics will be covered within the above categories.

Java on your own machine

If you have your own computer, and you don't have the Java JDK installed, you will want to get that so you can write and run Java programs.

You need the Java JDK (Java Development Kit), not simply 'Java'. There are lots of places online where installing this is explained.

Java development

In this module we will be using Linux, writing programs in a text editor, and compiling and running programs using the command line. If you have experience of an Integrated Development Environment (IDE) that you want to use (such as Eclipse) then you may do so, so long as you comply with the requirements of the module. The use of IDEs will not be supported on the module.

Exercises

On the sheet that you should have by now, there are some exercises to try. You will need to be able to log in to the machines to do the exercises.

You should be able to do some of the exercises, maybe most. One of them is a bit more advanced and may take more time. If you don't complete them all today then please try them in your own time.