1 Checkpoints

1.1 Introduction

This section contains details of what you are required to complete during this course. Read this **before** *you start the course.*

This programming course is defined and assessed via a set of *Checkpoints*, each of which involves at least one programming task. In order to complete these *Checkpoints* you will need to read and study the course notes. The laboratory is also supported by demonstrators who are there, and paid, to help *you* through it.

1.2 The Checkpoints

There are **5** compulsory *Checkpoints* and one *optional*. You work through the *Checkpoints* at your *own pace*. If you have had significant previous experience of programming, for example from *Computer Science 1Ah*, then you will find first three *Checkpoints* very simple. In this case you are expected to attempt the optional *Checkpoint* **6**.

The *Checkpoints* are:

1. ADDRESS PROGRAM: Extension of "Hello World". (5%)
2. VARIABLES AND ARITHMETIC: Two item checkpoint of basic input/output, arithmetic and variable types. (10%)
3. ROOT OF QUADRATIC: Calculation of root of a quadratic equation used to demonstrate conditional statements. (20%)
4. DAMPED HARMONIC OSCILLATOR: Calculation and display of amplitude of a damped harmonic oscillator under various damping conditions. (30%)
5. FAIR AND WEIGHTED DICE SIMULATION: Simulation to frequency histograms for the multiple fair and weighted dice (35%).
6. : Monte-Carlo simulation to calculate π, and the volume of a water molecule. (*optional replacement for checkpoints 1 to 3 for experienced programmers only, 35%*)

1.3 Checkpoint Submission Dates

You should get checkpoints marked by the demonstrators *immediately* after completion. You will typically get useful feedback from each checkpoint that will help you with the next. The **absolute** final sumbission dates for the checkpoints are:

- **Checkpoint 1-3 : 5.00 pm, Thursday 20th October**
- **Checkpoint 4-6 : 5.00 pm, Thursday 3rd November**

checkpoint submitted after these dates will **not** be counted towards the assessment of this course.

1.4 Checkpoint Grading

The first three checkpoints will be graded on a **3** point scale and the second three on a **5** point scale. The grading will include,

1. Function of the code, does it do what you think it does.
2. Design and layout of the code, including the use of good structure, comments and use of sensible variable names.
3. Understanding of the problem and the ability to answer questions regarding the program.

The final *mark* from *Scientific Programming* will contribute **10%** towards your total mark for the Physics 2A course.

1.5 Checkpoint Requirements

The programs submitted for *Checkpoint* **must** be your own work, and **must** not be copied, either in whole or in part from other students or directly from textbooks. Submitting or attempting to submit other peoples work as your own is a breach of *University Code of Student Discipline*. This does *not* mean you cannot seek or give assistance to your class-mates, but you must not give or copy programs.

On a less legalistic stand, remember *programming* is a very valuable skill, both as part of your degree and as a general, and very marketable, skill in future employment. (Look in any recruitment paper and see how many jobs require JAVA programming!) The only way to become a competent programmer is to sit in front of a terminal and "*write, test and debug programs*". This course gives this opportunity with demonstrator cover to assist you. Make the most of it, you *will need* the skills you acquire here!

What Next?

You are now ready to start the course. The next few sections will tell you how to log-in to the computer systems and the basics of UNIX. These sections are best read while sitting in-front of a terminal. If you want to "read ahead" and start the JAVA language, jump to section 6.