1 Introduction

Read this and the next section **before** *you attend the first class.*

1.1 Synopsis

The aim of this course is to teach the basics of scientific computer programming using JAVA in the Unix environment. This course is taught "on-line" in a series of **six** 3 hour sessions using the School's Computational Physics Laboratory and one EUCS Micro-lab. This course is taken in weeks 2 to 7 of Semester 1, starting on Monday 26th September, 2005.

1.2 Why learn a computing language

All computer programs, whether they perform simple calculations or run an entire network of systems, are written in a *programming language* of some type. Anybody wanting to use computers beyond simple WEB browsing, word processing and e-mail has to learn some type of programming language. Most scientists, and especially physicists, need to make more than this basic use of computers and so need programming skills; now is a good time to start!

1.3 Uses of Java

Say JAVA and what springs to mind is animation and user interactions on WEB pages using either,

1. JAVASCRIPT which is a small section of JAVA like code that is inserted into WEB pages to perform functions that normal HTML does not support or,

2. APPLETS being small self contained JAVA programs that are run by a net browser to perform animation, user interfaces and other complex operations.

However JAVA is much more than just a WEB graphical *add-on*, it is a fully functional general purpose computer language with excellent graphics and seamless integration with windows/menus and WEB environments. More importantly it is truly machine independent, so JAVA programs will run, without modification, on any system on which the language has been implemented. This makes it an ideal modern language in which to teach elementary computing.

JAVA is a very new language which is still developing. Numerical support is there but it not well developed in particular there are few, if any, numerical libraries at present. In addition most implementations of JAVA are slow and rather inefficient. However this is changing and we expect to see a rapid emergence of JAVA as a scientific computing language via such projects as JAVA GRANDE which aim to implement JAVA on massively parallel supercomputers.

1.4 What you will learn

This short course will concentrate on the programming language and you will be writing short *applications* to perform basic calculation and elementary simulation tasks. There will be sufficient graphics to make your applications usable. To allow this there must be an element of *black-box*, where at times you will have to use a recipe that works without fully understanding what it does.

You will be invited to explore more adventurous JAVA features, for example simple OBJECTS, which will be covered in more detail in COMPUTER SIMULATION, optional course in semester 2, or COMPUTATIONAL METHODS in Junior Honours.

1.5 Other Languages

There are a vast range of computer languages, the most used being:

- "C" is powerful general purpose language used extensively throughout the software industry for the last 10 to 15 years. Used extensively in system codes and many current applications, but most new applications are being written in one of the newer alternatives.
- C++ is a super-set of "C" which adds object-oriented programming. Good modern language currently much favoured by the software industry. It lacks standardised graphical and network, is rather syntactically difficult and the resultant programs are frequently difficult to understand and maintain.
- FORTRAN is the traditional numerical language with excellent numerical libraries and support for parallel computer systems. There is an optional course at SENIOR HONOURS level of using FORTRAN-95 with parallel computing extensions for large scale computational modelling.
- PERL. Modern interpreted language with "C++" like syntax with very powerful character and text manipulation features. It is widely used for system control programs but is currently neither numerically efficient enough nor has sufficient numerical features to make it a viable scientific computing language.

The current *language of favour* for code developers appears to be C++, however since JAVA has all the (useful!) features of C++ with integrated graphics, networks ,WEB, multi-media, machine independence and simpler syntax it is well placed to take over. Hence you are going to learn JAVA.

1.6 Flavours of JAVA

The main developer of JAVA is Sun Microsystems Inc. This course has been developed using Sun Microsystem JDK 5.0, currently at update 4, which can be downloaded from `http://java.sun.com` with a free user licence[1].

There are other flavours of JAVA notably from IBM plc which does not currently implement the JDK 5.0 features used in this course and gnu who are developing a open source JAVA compiler as part of the gcc compiler. This does not currently support JDK 5.0 features nor has fully functional graphics.

[1] Subject to the restrictions detailed in the Binary Code Licence Agreement

1.7 Documentation

This course is available on-line at:

```
http::/www.ph.ed.ac.uk/˜wjh/teaching/Scientific-Programming/
```

which also includes a set of complete program Examples many of which are useful starting points for the checkpoints. These examples are available as CLICKABLE LINKS form the on-line version of this booklet.

These pages also contains links to many other on-line documents, including the details of the local classes all the full JAVA on-line documentation set.

There are relatively few JAVA books that teach “basis programming”, most either assume that you are an experienced “C” programmer and want to convert to JAVA, or that you want to use JAVA primarily for WEB animations. Neither are appropriate to you.

The three most useful book appear to be:

1. JAVA GENTLY FOR SCIENTISTS AND ENGINEERS Judith & Nigel Bishop, Addison-Wesley.
   Good simple book with physicist in mind. Uses its own classes. ≈£37.00.

2. Small Java: How-to-program by Deitle and Deilte. Deitel & Associates Inc Pub.
   short(er) version of the full book, see below, with little on graphics or user interfaces, but sill 600 pages. Starts as the very beginning and explains every step in detail. ≈£43.00.

3. Java: How-to-program by Deitle and Deilte. Deitel & Associates Inc Pub.
   Very very long detailed book (1,500 pages). It starts at the very beginning and explains every step in detail, including graphics and user interface. ≈£54.50, but available second hand from about £30.00.

There are many books on JAVA being published every month, the best way to find the book you like is to browse the shelves of the major city bookshops.