F20RO/F21RO INTELLIGENT ROBOTICS
Notes on Java for Coursework 1b

These notes are intended to give you sufficient information to edit, compile and execute the Java program, `robot.java`

The `robot.java` program has been annotated with various comments to help you identify key sections. The comments are identified as "Note from Nick Taylor". The most important such annotation indicates that the method ***`calcThetas()`*** is where you are expected to enter your control equations to calculate the four theta angles from the *x*, *y* and *z* co-ordinates and orientation angle, *alpha*, supplied by the user through the GUI.

You may use any development environment that you are comfortable with as long as it supports Java JRE 1.8. We recommend forking the *gitlab-student* project –
https://gitlab-student.macs.hw.ac.uk/f21ro2021-2022/intelligent-robotics-cw-1b
and using an IDE like Eclipse or compiling and executing from the command line on the MACS Linux Virtual Machine or a computer in the MACS Linux Lab in Edinburgh (EM2.50). If you are not using one of these methods then you will require the *jars* here –
https://www.dropbox.com/sh/t4db9vwr2o5sbsj/AACmDJFM8fl_F6UlfjQl_tFda

This video –
https://web.microsoftstream.com/video/97371b7f-8085-4012-9fa1-ec3ee3f4f0a7
explains how to import the program from *gitlab-student* to Eclipse with the correct Java JRE (1.8). If you are using the MACS Linux Virtual Machine or a computer in the MACS Linux Lab in Edinburgh then you could simply compile your program from the command line using the command –

```
javac robot.java
```

This will create a file in your home directory called `robot.class` and this is the file which you should execute to run the program. You can run it with the command –

```
java robot
```

Note that the code which you are required to write should only differ from C++ (or, indeed, C) in the names given to the mathematical functions. The Java definitions of some mathematical functions which you may require are –

```
Math.cos(x)         Returns the cosine of x radians
Math.sin(x)         Returns the sine of x radians
Math.atan2(x,y)     Returns the arctangent of x/y
Math.sqrt(x)        Returns the positive square root of x
```

Note that the parameters and results of these functions are all of type `double`.
Doubles can be converted to floats in the same way as in C and C++

```
myFloatVariable = (float) myDoubleVariableOrValue;
```

The program also defines the function `sqr(x)` to square a `float` passed as its parameter.