Lab Environment

Christophe Rhodes

Motivation

- Describe the lab environment clearly
- Allow you to recreate it on personal computers

Ingredients

1. Compiler and runtime
   - Java
   - C++

2. Build tool
   - Make

3. Testing framework
   - JUnit 4
   - cppunit

4. Version control system
   - git

5. Text editor (or IDE) of your choice

Installation: Linux (Debian/Ubuntu)

```
apt install <all the things>
```

1. Java:
   - default-jdk, default-jdk-doc, default-jre
   - make, make-doc, git

2. C++
   - g++, gcc-doc
   - libcppunit-dev, libcppunit-doc
   - make, make-doc, git

Installation: Windows (Java)

1. install msys2
2. run msys2 msys and install packages (`pacman -S`):
   - `make`, `git`
3. install Java (the latest JDK) from Oracle

Installation: Windows (C++)

1. install msys2
2. run msys2 msys and install packages (`pacman -S`):
   - `mingw64/mingw-w64-x86_64-gcc`
   - `mingw64/mingw-w64-x86_64-cppunit`
   - `make`, `git`

3. always run lab code using the MinGW 64-bit executable – it won't work, with confusing errors, if you use MSYS

4. but: always update your MSYS system using the MSYS executable

Installation: OS X (C++)

1. install xcode developer tools
   - `xcode-select --install`
2. install homebrew
   - sorry, you're on your own here
3. install cppunit
   - `brew install cppunit`

Installation: OS X (Java)

1. install xcode developer tools
   - `xcode-select --install`
2. install Java (the latest JDK) from Oracle

Work

1. Follow the instructions for your operating system to install the lab environment on your own computer
   - any problems: ask for help on the forum
2. Test your installation
   - start the environment
   - `git clone`
     `http://gitlab.doc.gold.ac.uk/crhodes/is52038b-labs`
   - `cd is52038b-labs/01/<lang>`
   - `make test`
   - Read the output carefully.
3. Select your programming language for labs and assignments from the choices provided on learn.gold.