



Programming 03

01a - Introduction

Administrivia

- Dr Andy Guest
- a.guest@yorks.ac.uk
- Tel – (01904) (62)6524
- Keep Checking
 - e-mail
 - moodle

Module Goals

- To gain a good understanding of:
 - How memory works in a computer
 - How operations are carried out at a low level
 - The basics of C programming
 - The basics of Assembly Language
- To develop your skills in
 - Analysing a problem and designing a solution
 - Coding

Programming

- Programming is a **knowledge** and a **skill**
- **Knowledge**
 - from these lectures, from books, etc.
- **Skill**
 - from coding, coding and more coding

Developing Software

- Consists of two skills
 - Breaking a problem down in to small steps
 - Coding those small steps
- Last year you had to learn both in learning Java
- This year you only need how to code in C and Assembly language!
- Above is an over-simplification but once you've learned one language, all the others are easier to learn

Module Outline

- **Memory**

- How memory is used
 - The Stack, Frames, The Heap
 - Primitive Variables Vs Objects
 - Memory at a very low level

- **C Programming**

- Introduction
 - Origins, Uses, Differences to Java
 - The Basics
 - How to do the stuff you know how to do in Java but in C (except for classes and objects because C doesn't have those!)
 - The Joy of Pointers

Module Outline

- **Inside The Chip**
 - Processor, Registers, The Stack
- **Assembly Language**
 - Introduction

Schedule

1. Memory 1, Introduction to C 1
2. Memory 2, Introduction to C 2
3. C Basics 1
4. C Basics 2
5. The Joy of Pointers 1
6. The Joy of Pointers 2
7. Inside the Chip, Introduction to Assembly
8. Assembly Language
9. Graduation Week – No classes
10. Assembly Language
11. Assembly Language
12. Loose Ends, Reflection, Assessment Work

Assessment

- Portfolio
 - Submitted at the end of the module
 - Consisting of
 - C Exercise
 - Document describing your solution to the exercise, explaining how it works and how and why you use pointers (25%)
 - C Code solution to the exercise (25%)
 - Assembly Exercise
 - Document describing your solution to the exercise, explaining how it works and how data moves through registers and the stack (25%)
 - Assembly language solution to the exercise (25%)
 - Submitted through gitlab (code and documents)

Assessment Notes

- I want to see you understand what is happening at the low level in the exercises
- The documentation gives you chance to describe how to solve the exercises even if you can't quite manage to put it into code

Tools

- Command Line Tools
 - We will be using Linux command line tools to compile, debug and execute code
 - No pretty GUIs!
- Text Editor rather than an IDE
 - If I'm feeling sadistic I might even introduce you to VI based editors!!

GitLab

- Online Source Code Management
- Version Control
- Hosted on university servers
 - <https://git.yorkdc.net:8888/>
 - User name and password same as computing network account
- GitHub Guides
 - <https://guides.github.com/>