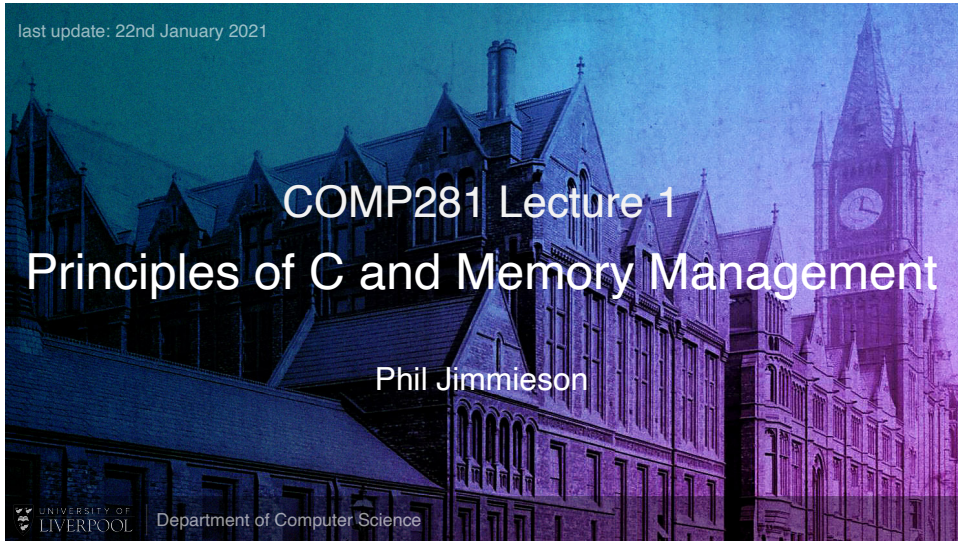
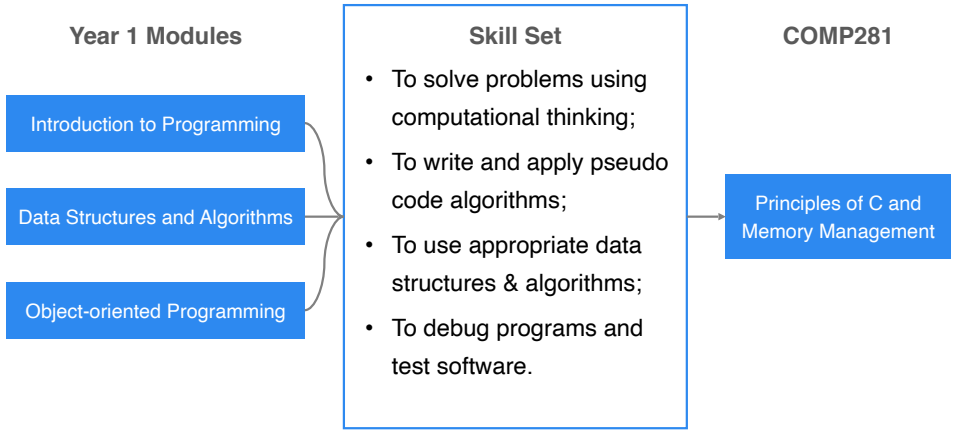
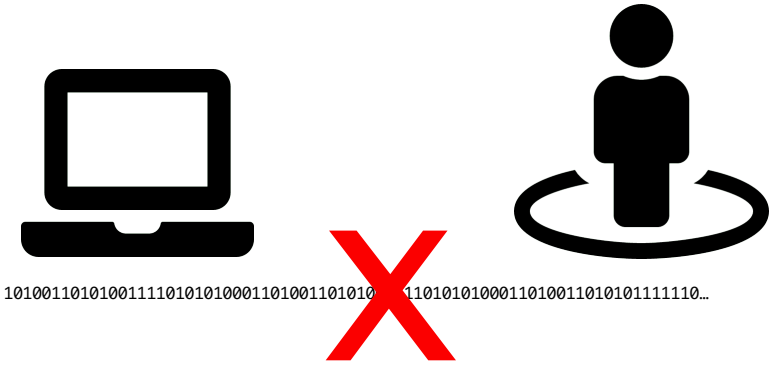


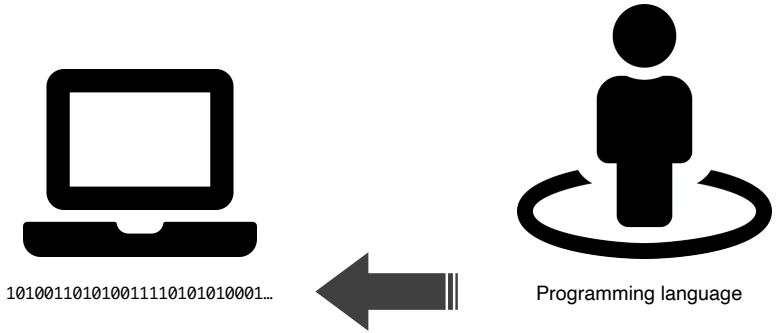


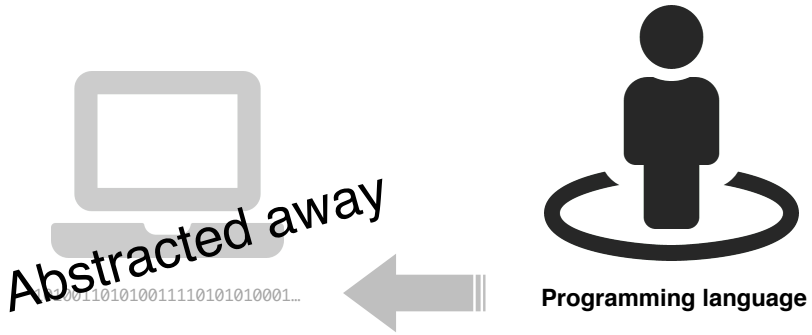
Today

- COMP 281 Principles of C and Memory Management?
  what this module is about
- General module information.

What is C?







The “Hello, World!” program

```c
#include <stdio.h>
int main(void)
{
    printf("Hello, World!\n");
    return 0;
}
```

code

compile

output

Executable Program

Edit -> compile -> run

- Make this into a file called `hello.c` using a text editor, e.g., vim
- Compile into a program and run:

```
% gcc hello.c
% ./a.out
Hello, world!
%
```

- ☺

C programs are built up of functions

```
#include <stdio.h>
int main(void)
{
    printf("Hello, World!\n");
    return 0;
}
```

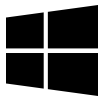
- Functions
  - Take in arguments
  - Compute something
  - Return a result
- The `main()` function
  - is where a program execution starts

Why do we study C?

C History

Year		Designed by
1960	ALGOL	International Group
1967	BCPL	Martin Richards
1970	B	Ken Thompson
1972	Traditional C	Dennis Ritchie
1978	K&R C	Brian Kernighan & Dennis Ritchie
1989/1990	C89/C90	ANSI&ISO Committee
1999	C99	ISO Committee
2011	C11	ISO Committee
2018	C18	ISO Committee

C powers the World – Operating Systems and Databases


Windows / Windows Phone


OS X / iOS


Android


Linux

Oracle Database


MS SQL Server


MysSQL


PostgreSQL

C powers the World – Embedded Systems


Plane


Car


Microwave


Dishwasher


Drone


Vacuum Robot


Coffee Machine


Smart Watch

C Advantages – Portability and Efficiency

- C is almost a ***portable assembly language***. It is as close to the machine as possible while it is almost universally available for existing ***processor architectures***.
- Compilers, libraries, and interpreters of other programming languages are often ***implemented in C***. Interpreted languages like *Python*, *Ruby*, and *PHP* have their primary implementations written in C.

C Advantages – System Resource Management

- Arbitrary memory address access and **pointer arithmetic**.
- **Deterministic usage** of resources that fit for resource-limited systems.
- C has a very **small runtime**. The memory footprint for its code is smaller than for most other languages, e.g., C++.

C Advantages – Lingua Franca

- Many implementations of **new algorithms** in books are first (or only) made available in C by their authors.
- C is an old and widespread language – it's easy to find **all kinds of algorithms** written in C.

Anything bad about C?

Disadvantages

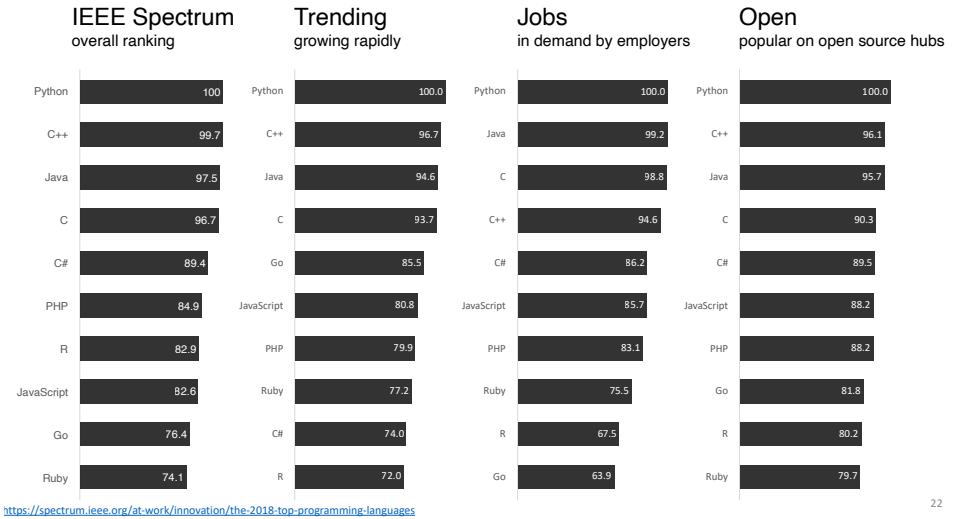
- No concept of Object Oriented Programming (OOP).
- No concept of Namespace.
- No Constructor or Destructor.
- Difficult to debug.
- Compilers cannot handle exceptions (run-time errors).
- No strict data type checking, e.g. an integer value can be passed for floating datatype).

Programming Language	2021	2020	2019	2014	2009	2004	1999	1994	1989
C	**1**	**2**	**2**	**1**	**2**	**2**	**1**	**1**	**1**
Java	2	1	1	2	1	1	14	-	-
Python	3	3	4	7	5	10	20	21	-
C++	4	4	3	4	3	3	2	2	3
C#	5	5	6	5	7	8	30	-	-
Visual Basic .NET	6	6	5	11	-	-	-	-	-
JavaScript	7	7	8	8	8	7	19	-	-
PHP	8	8	7	6	4	5	-	-	-
SQL	12	10	9	-	-	6	-	-	-
Ruby	15	11	10	10	10	22	-	-	-
Objective-C	18	13	11	3	40	45	-	-	-
COBOL	31	??	25	20	15	11	3	9	18
Lisp	36	??	28	13	16	14	8	6	2
Pascal	??	??	199	14	14	96	5	3	14

https://www.tiobe.com/tiobe-index

Programming Language	2021	2020	2019	2014	2009	2004	1999	1994	1989
C	1	2	2	1	2	2	1	1	1
Java	2	1	1	2	1	1	14	-	-
Python	3	3	4	7	5	10	20	21	-
C++	4	4	3	4	3	3	2	2	3
C#	5	5	6	5	7	8	30	-	-
Visual Basic .NET	6	6	5	11	-	-	-	-	-
JavaScript	7	7	8	8	8	7	19	-	-
PHP	8	8	7	6	4	5	-	-	-
SQL	12	10	9	-	-	6	-	-	-
Ruby	15	11	10	10	10	22	-	-	-
Objective-C	18	13	11	3	40	45	-	-	-
COBOL	31	??	25	20	15	11	3	9	18
Lisp	36	??	28	13	16	14	8	6	2
Pascal	??	??	199	14	14	96	5	3	14

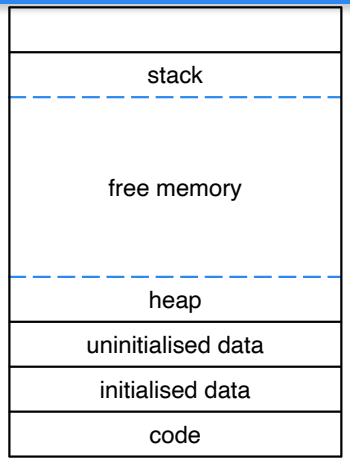
https://www.tiobe.com/tiobe-index



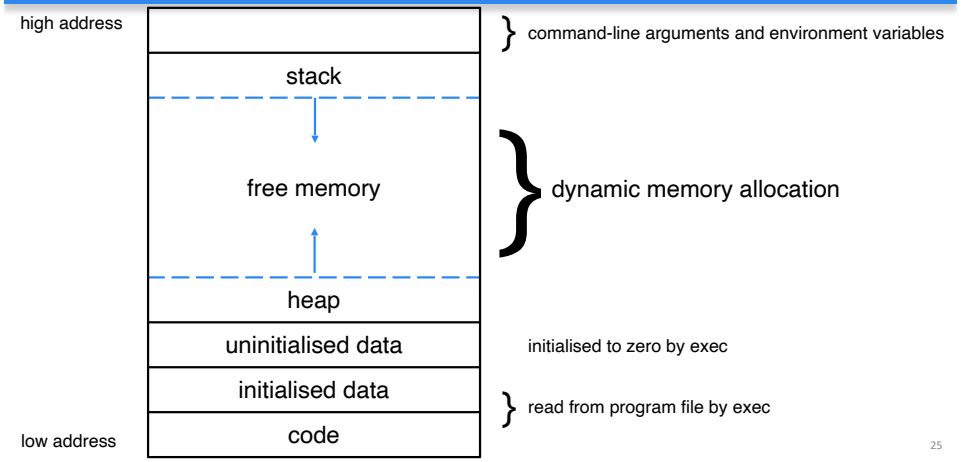
https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages

What's Memory Management, and Why do we study it?

Memory Organisation of a typical program



Memory Organisation of a typical program



COMP281

Principles of C
+
Memory Management

- You already have experience with programming, so hopefully you can pick up C syntax quickly and start having ***fun*** with coding.
- You will further practice programming & debugging stills.

General Module Information

Contact

Phil Jimmieson

phil@liverpool.ac.uk

0151 795 4236

https://cgi.csc.liv.ac.uk/~phil

Room 1.20 Ashton Building

Module Delivery

- The equivalent of two lectures per week
  - Start in Week 1
  - Asynchronous delivery - accessed from the Canvas site
- Two-hour lab session per week (synchronous delivery)
  - Start in Week 2
  - You have been assigned to a group in one of four slots:
    - Wednesday 9:00 - 11:00 Group 4 & 11:00 - 13:00 Group 1
    - Thursday 13:00 - 15:00 Group 2 & 15:00 - 17:00 Group 3

Learning Outcomes

At the end of the module, you will be able to

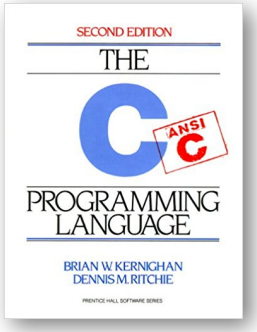
- Analyse and explain the use of memory resources within software applications, including memory usage on the **stack** during function calls and **heap**-based dynamic memory management;
- Use debugging tools to **inspect memory usage**, and to assist in the development of software;
- Develop applications using the C programming language, including use of **command-line** driven C development tools;
- Deal with underlying memory-based issues in using **dynamic data structures** through the implementation and management of at least one familiar data structure using the C programming language.

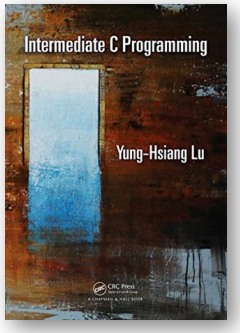
Assessment

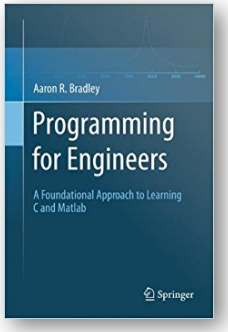
- The module is 100% CA
  - No exam
- There will be 2 assessments worth 50% each
- Schedule (current plan)
  - Assignment 1: set week 2, deadline week 4
  - Assignment 2: set week 4, deadline week 6

Module Texts

- There is no required text for the module. Recommended texts:







Setup Programming Environment

- The programming environment is setup on the lab machines in CS
- You should also set it up on your own machine
  - **Text editors**

Vim Sublime Text atom Brackets Visual Studio Code BBEdit

  - **The compiler**
    - Windows / Unix / Mac

The COMP281 Webpage

- Canvas web site: COMP281-202021
  - All lecture notes will be posted early in the week
  - Worked solutions for all assignments on the module

What you should be doing

Teaching Schedule

“

- Lectures: 2 one-hr lectures per week for 5 weeks -> 10 hr in total
- Labs: 1 two-hr lab per week for 5 weeks -> 10 hr in total
- Self study: 55 hours
- In total: 75 hours

”

- Module Specification

What you should be doing

Of course

- Watch the lectures
- Attend labs

Self study

- Practice is the key to study coding
- Try to do a bit of coding every few days
- Submit assignments on time

How to get help

If you are stuck during the module

- Ask questions by email
- Attend the lab session and ask demonstrators
- Check the worked solutions on the website
- Read the text/slides
- Google/Stack Overflow (but beware of PLAGIARISM!!)

Summary

Today

- Principles of C and Memory Management?
  what this module is about
- General module information.

After this lecture

- Setup your own C programming environment

Next lecture

- Compiling and Running C Programs
- C Language Basics