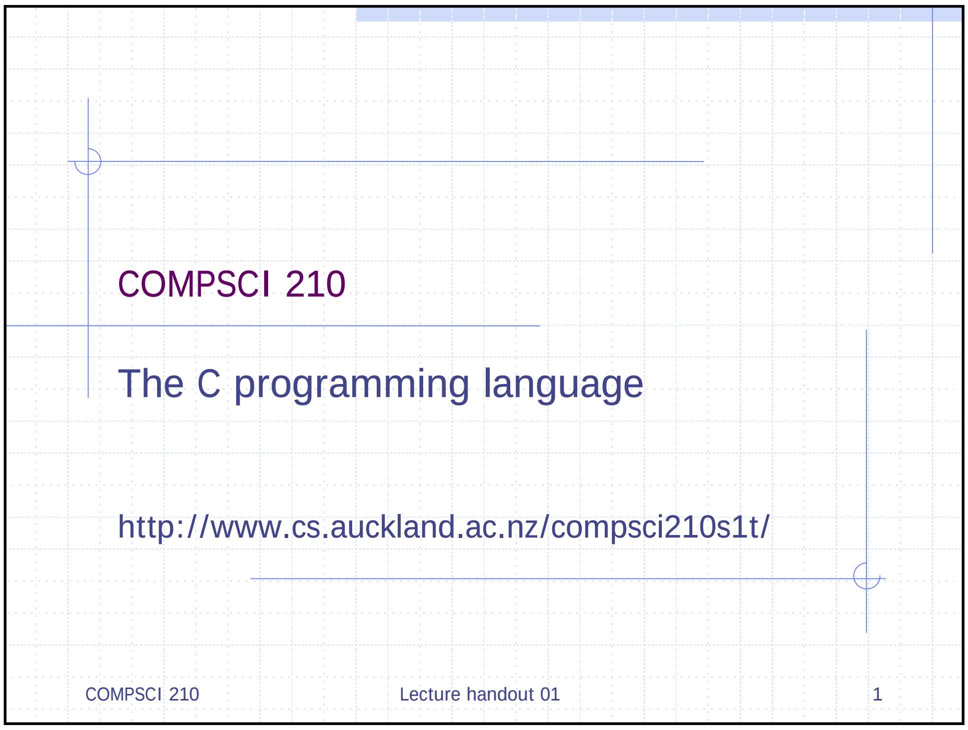
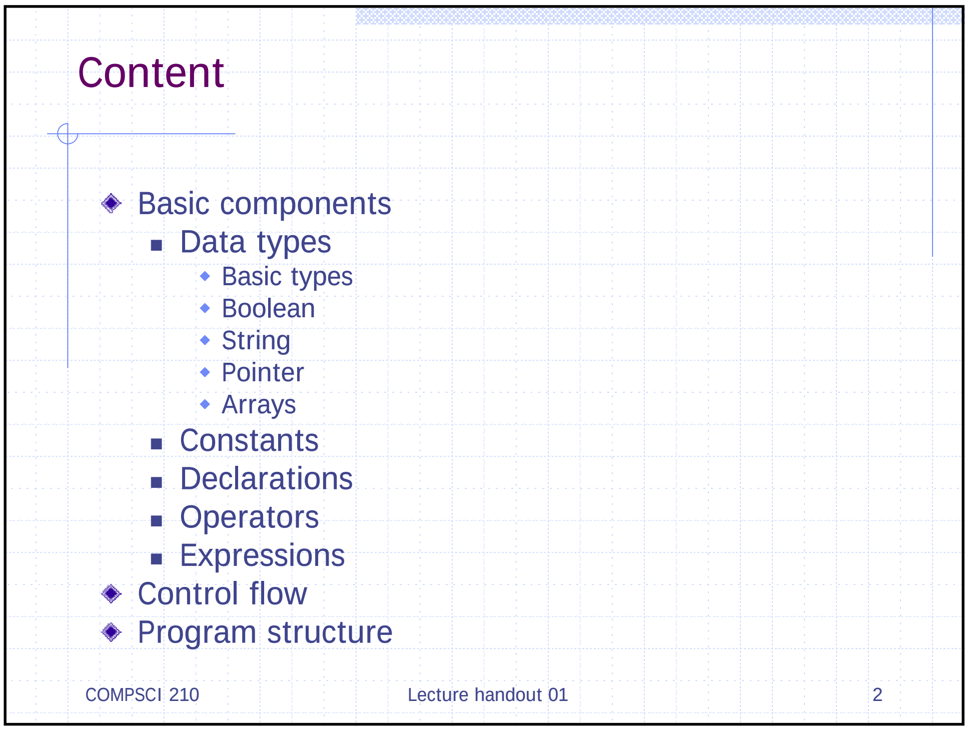


COMPSCI 210

The C programming language

http://www.cs.auckland.ac.nz/compsci210s1t/



Content

- Basic components
  - Data types
    - Basic types
    - Boolean
    - String
    - Pointer
    - Arrays
  - Constants
  - Declarations
  - Operators
  - Expressions
- Control flow
- Program structure

References

- Kernighan and Ritchie: The C Programming language
- Kelley, A, and Pohl, I, A Book on C: Programming in C, Addison-Wesley.
- Harbison, S, and Steele, G, C: A Reference Manual, Prentice Hall.
- 210 Lecture notes by Dr. Bruce Hutton
  - http://www.cs.auckland.ac.nz/compsci210s2c/lectures/

C Compiler

The assignment will be run using the C compiler (“**cc**” or “**gcc**”) on our LINUX machine **chaos.cs.auckland.ac.nz**.
  - Try to ensure that you use only standard ANSI C features
  - The compiler does not generate either error or warning messages.

Your source files names should be of the form “*.c” to compile C, not C++ code.
  - On many UNIX systems, you can use the flag “**-ansi**” to restrict the compiler to ANSI features.
  - The flag “**-Wall**” causes the compiler to display all warning messages.
  - Before compiling, check the compiling options (man gcc)

A C compiler is provided for free on any LINUX, UNIX, or MacOS X system.
  - It can be accessed as the “cc” or “gcc” command, via a terminal window.

- The URL http://www.thefreecountry.com/compilers/cpp.shtml provides information on many free C compilers.
- Install Cygwin on a Windows system.
  - This gives you a UNIX like system running on top of Windows, and includes the GNU C compiler, “gcc”.
  - The Bash shell provided by Cygwin is far superior to the DOS like command language (cmd) normally provided on Windows.
  - Information on Cygwin, look at the appendix of BH’s Unix lecture notes.
- Download MS Visual C++ Express edition
  - http://msdn.microsoft.com/vstudio/express/visualc/download/
- A commercial version of Visual Studio can be obtained from the University web site, https://www.se.auckland.ac.nz/msdn/.
- Bloodshed, provides a free C compiler, and is available from http://www.bloodshed.net/devcpp.html.
- Assignment next week on compiling C code

C: a bit of history

First generation computer languages: Machine code.

- Computer hardware is designed to understand and execute what is called “machine code” (instructions to tell the computer what to do). Different kinds of computer understand different machine code.
- a computer program: the bit pattern of the machine code to be loaded into the computer memory.
- The bit pattern could be specified manually, using switches on the front panel for the computer
- permanent storage (e.g., paper tape or magnetic tape), and loaded by a very simple “bootstrap” program.

Second generation computer languages: Assembly language.

- Specifying a computer program as the bit pattern of the machine code was very time consuming, and error prone.
- A human readable/writable form of machine code was developed, namely “assembly language”.
- Another program, called an assembler, was developed to take a textual version of the machine code, and translate it into machine code.
- Much easier than specifying the machine code.

Third generation computer languages: High level languages.

- Assembly language is very low level, and depends on the computer architecture.
- High level languages were developed, that were relatively machine independent, and more like the notation of mathematics. Fortran (~1955) and Basic (~1964): still had to build control statements out of one line if statements and goto statements, and there were no recursive functions.
- Pascal (~1970) and C (~1972) were developed, with support for data structures.
- Modern languages, such as C++ (~1983) and Java (~1995), are object oriented languages. A program, called a compiler, is used to translate the high level language into assembly language or directly into machine code.
- Third generation languages represent the main programming languages in use today.

C: what it does/does not (1)

C++/Java:

- Allow data and code to be grouped together into classes. Large class libraries are provided.
- New classes can be built by extending previously defined classes.
- Provide automatic garbage collection.
  - The programmer can allocate space for data structures (e.g., using “new” in Java).
  - When they no longer refer to the memory allocated, the “memory management system” has to work out that the memory can now be deallocated, and used for some other purpose.

C: what it does/does not (2)

C is a block structured language, with notions of arrays, records and pointers

- C supports control statements, such as for, while, if, and switch statements.
- Supports recursion, and the equivalent of value and reference parameters.
  - A value parameter is one where only the value of an expression is passed to a function.
  - A reference parameter is one where the address of memory is passed to a function, and the function can alter the contents of the memory referenced

C has no notion of packages or classes

- It is possible to split C programs into multiple files, but there is no notion of packages

C does not support overloading

- You cannot have variables with the same name as functions, or multiple functions with the same name.
- However, structure names are in an independent **scope** from other names, as are label names.

C is like a high level assembly language

- Almost anything that can be written in assembly language, can be written in C.
- Partly due to the fact that the way code and data is viewed in C is very close to the way it is represented at the machine level.

C: memory allocation

C lacks garbage collection

- In C, a variable declared of array or structure type contains the array or structure value itself, not a pointer to an array or structure.
- Space for the array or structure is allocated as a part of the declaration, so there is no need for dynamic memory allocation.
- The array size must be a compile time constant, so the compiler knows how to lay out memory.

In Java, a variable declared of array or class type is a pointer.

Space is not automatically allocated for the array or object

Space must be explicitly allocated using “new”.

Garbage collector determine which blocks of memory are not longer accessible via any variables.

The memory will be de-allocated by the garbage collector, to become available for re-use.

In C, It is possible to declare a pointer variable (initially null), that can be used to point to a block of memory.

Memory may be dynamically allocated by the malloc() function.

No support for automatic garbage collection.

The programmer must determine when the memory is no longer in use, and explicitly deallocate the memory (using free()), when they want to make it available for re-use).

Error prone: you must check that each malloc instruction is ended by a free instructions.