How it all works

Edit, Compile, Link, Execute

Tools you need to program

- A programming language
- An editor to create source files
- Compiler to create object files
- Linker to create executable files

C Programming Language

- Consists of
  - Keywords
  - Functions
  - Operators

32 C keywords

auto (obsolete)	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto (don't use)	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

Functions

- We use functions to do most of the work
- printf to print to screen for example
- puts is another print function (put string)
- key = getchar() will get a character from the keyboard
- root = sqrt(27);
- We need to use functions to make our programs useful. Later we will write our own.

Operators

- = assignment
- +, -, *, /
- % modulo division (remainder)
- ==, >, >=, <=, <, != (relational)
- &&, ||, ! (logical) (and, or, not)
- ++., -- increment and decrement by 1
- +=, -=, *=, /=, %= (assign and math)

Values and Variables

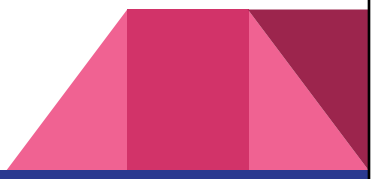
- We can have strings in our code like “Hello World”
- Numbers like 27
- “Variables” are what we use to store values we don’t know when we are writing the code.
- Same concept as x and y in algebra. Placeholders that hold an unknown.

Syntax

- It’s the way the language is put together.
- English has a looser syntax: we can still understand Yoda, mostly
- Programming languages follow an exact syntax.
- A “syntax error” is when the computer doesn’t understand what we are saying.

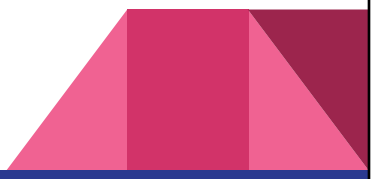
Creating a Program

1. Create a Source File
2. Compile
3. Link
4. Execute



Step 1 – Create a Source File

- We use a text editor to write programs
- C programs have file extension .c
- 'C' is case-sensitive so we have to careful with upper and lower case.
- The file we create is known as "source code"



Step 2 - Compile

- The compiler is a program
- It converts source code to object code.
- First it looks at precompiler directives these are our include files.
- The compiler inserts the include file into our source file. (without modifying the file)
- Syntax errors are checked for, and reported if found.
- If successful an object file is created. (.obj)

Step 3 - Link

- The linker builds the program file (.exe)
- It takes binary (machine language) files, the object files and needed library files and creates an executable file.
- Library files have the code for our functions like printf.
- You will get linker errors if a function is not found. Otherwise an executable file is created.

Step 4 – Run

- After successful linking you will have a .exe in your directory which can be run.
- You can now test your program for logic errors.
- Logic errors happen when the program does what you tell it do to but it's not what you wanted.
- Correcting logic errors is called debugging.

Two Main Types of Errors

- Compile-time
  - Syntax errors (generally easy to correct, although can be frustrating with 'C')
- Run-time
  - Logic errors (the program doesn't work as it's supposed to).
  - Usually more difficult to find.
  - With 'C', fewer problems than you might expect are caught at compile time which can make debugging more difficult.