1 Anatomy of a C program

1.1 Programming languages

- The main processor in a computer works with *machine code.* The computer is only able to handle programs written in machine code.
- It is almost impossible to write machine code.
- *Assembly language* is somewhat easier to write, but still very difficult; it is very close to the machine code.
- Assembly language was used when very efficient code was wanted.
- The language C is used almost universally to produce efficient code much more easily than assembler language. Also, all machines have the same kind of C, and wildly differing machine codes.
- You write a program in C, such as `hello.c` below, use a *compiler* such as `gcc` to translate it into machine code. By default, the machine code is stored in a file called `a.out`.

The single command `a.out` will cause the computer to execute the program. Here is a Hello, World program in C.

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

- **What printf() does.** The `printf()` statement prints the message on the terminal (screen, monitor). This action is called **output.**
- **Need for #include <stdio.h>** The statement `printf()` is not 'part' of the C language; it is a separate routine whose general properties are in the file `stdio.h` which is stored in some recognised place in the computer. The `#include` statement is necessary; otherwise `gcc` will not recognise the `printf()` statement.

  The file `stdio.h` is called a 'header file.' Hence the suffix **.h**.
- **Need for \n** This pair of symbols defines a carriage return. Without the carriage return, your output will get mixed up with the next line.
- **The main program.** The real business of the program is in the

  ```
  main ()
  { .... }
  ```

  Every C program must contain this — called the 'main routine.'