What C Programming

- Structured programming
- Highly portable
- Fast compilation and execution
- Extended with build-in functions
- Only 32 C Programming keywords

Structure

- Comments
- Preprocessor directives
- Global declarations
- Main function
  - Local declarations
  - Executable statements
  - Return statement
- User-defined functions

C Character Set

- Alphabet
  - Uppercase: A-Z
  - Lowercase: a-z
- Digits
  - 0-9
- Special characters
  - - ~ ‘ ! @ # % ^ & * ( ) _ - + = | \ { } [ ] : ; " ' < > , . ? /
- White space characters
  - Blank space, new line, horizontal tab, carriage return, form feed

Identifiers

- User-defined names given to variables, functions and constants
- Case sensitive (varName does not equal varname)
- Can contain, but not begin with, a digit (0-9)
- Cannot contain special character other than underscore (_)
- Cannot use C Programming Keywords
- Identifiers should be descriptive, meaningful, and unique

Printf function

- printf() is a built in C function used to perform output operations such as displaying data on the screen
- printf() is imported using the <stdio.h> header, which stands for standard(std) input(i) output(o)

printf(“Welcome to C Programming”);

Output: Welcome to C Programming

Format Specifiers

- Format specifiers represent the value of a variable within a string. For example: printf(“string”);
- Both the data-type of the variable and the format specifier must match
- Format specifiers and variables are organized from left to right, so to output var_a, then var_b, then var_c:

pritnf(“%d %d %d”, var_a, var_b_, var_c);

Escape Sequences

- Escape sequences are used to signify characters which cannot be typed in a string without breaking the code
- A double quotation in a printf statement for example would end the string too early:

~~printf(“My double quotation ” here”);~~

printf(“My double quotation \” here ”);