

CO650 Advanced Programming

C++ Basics and Pointers

Basics

The main Function

- The entry point for C++ programs
- Invoked by the operating system
- Main returns an integer value. 0 indicates the program terminated normally and any other value indicates an error. How the return value is handled is Operating System dependent.

#include

- A mechanism that allows code located in one file to access code located in other files.
- A preprocessor directive that precedes a file name
- The contents of the included file are treated as if they were embedded within file in which the directive is placed

```
#include <path_filename>
#include "path_filename"
```

- <> search for the file in the directory designated as the include directory within the IDE
- "" instructs processor to look for file in the same directory as the source file containing the directive and if not present the designated IDE include directory
- "" if a full path is specified then the preprocessor will only look in that directory

#include header files

- Some files may require code that is in other files in the project. In this case you must explicitly indicate that a file requires the code in a header file.
- The include directive should be invoked within the file that requires the access to the definitions within the header file.

namespaces

- An identified declarative region that avoids naming conflicts
- Identifiers are accessed using the scope operator ::
- The using keyword avoids having to scope each object within the namespace

Addresses

- The computer's memory consists of memory cells in which the variable's value is stored.
- Each memory cell has a unique address
- The cells are given consecutive addresses

- The data value stored in the address is known as the **rvalue**
- The address itself is known as the **lvalue**.
- The OS decides at what address the variable will be stored when the variable is defined
- We can access this address through the **reference operator &**.

Pointers

- A pointer is a variable that contains the memory address of a variable.
- Pointers are defined using the asterisk symbol.(*)
- The type of the pointer is the type of the variable the pointer points to.
- After definition, the pointer does not contain a valid address. Its contents are undetermined.
- A Pointer should always be initialized.
- If the address is unknown at definition, assign 0 as there is no address 0;
- When declaring more than one pointer of the same type, repeat the *

```
int *a, *b *c
```

- Through the pointer we can access the value within the variable whose address is stored in the pointer
- The **dereference** operator * precedes the pointer to return the variables value
- The dereference operator translates to “value pointed by”

Reference

- An alias for a variable, through which the data within the variable can be modified.
- The reference holds the address of a variable
- The reference is declared using the reference operator &
- The reference must be initilized during definition

```
type& referencename = value;
```

- Once initialized the reference address can't change (unlike a pointer)
- After declaration the reference can be treated as a normal variable