Chap 0: Overview

- Overview of basic C++ syntax
- Refresh programming basics
- C++ Vs. Java differences
- Coding conventions used
Basics - 1

• Comments
  – single line: //
  – multi-line: /* ... */
• Identifiers and keywords
  – {__a-zA-Z}{{_a-zA-Z0-9}*}
  – keywords are reserved words
• Fundamental data types
  – bool, char, int, float, double
  – modifiers: signed/unsigned, short/long
Basics - 2

• Variables
  – double radius; int count, i;

• Constants
  – literal: ‘A’, ‘2’
  – named: const double PI = 3.14159;

• typedef statement
  – typedef double Real;
Basics - 3

• Assignments and expressions
  – arithmetic expressions
  – relational and logical expressions
• Implicit type conversion
  – automatic type conversion with no loss of precision
• Explicit type conversion
  – static_cast<type>(expression)
  – int ivol = static_cast<int>(volume);
Basics - 4

• Input
  – int a; cin >> a;
  – int a; scanf("%d", &a);

• Output
  – cout << "Output is: " << a << " \n";
  – printf("Output is: %d\n", a);
Basics - 5

• Functions
  type name (formal argument list)
  {
    body
  }

• Selection statements
  – if, if-else, if-elseif-else, switch-case

• Iteration statements
  – while, do-while, for
  – break, continue
Basics - 6

• Arrays
  – one dimensional: int arr[100]; arr[i] = 10;
  – multi-dimensional: int arr[100][10]; arr[i][j] = 10;
  – arrays are passed to functions by reference

• C++ strings
  – string str = “EECS 268”;
  – size(), length(), compare, concatenate, index, etc.

• C strings
  – char str[100];
  – Null character ‘\0’ terminates the string
  – strlen(), strcpy(), strcmp(), strcat(), index, etc.
• Structures – to group data items

```cpp
struct Person{
    string name;
    int age;
    double gpa;
};
```

```cpp
struct Person students[100];
students[0].name = "Adam Smith";
students[0].age = 20;
```

– structs are passed by value to another function

see A1-CppJava.cpp
Basics - 8

- File input / output – provide persistent storage
  - ifstream, ofstream, fstream

```cpp
ifstream inFile;
inFile.open("file.txt");
inFile >> ch;
inFile.get(ch);
ch = inFile.peek();
inFile.ignore(n);
```

```cpp
ofstream ofile;
ofile.open(filename);
ofile << ch;
ofile.put(ch);
ofile.open("file", ios::app);
```
C++ (Compared to Java)

- *structs* used to group data variables.
- C++ uses preprocessor for macros, file-inclusion.
- C++ can have stand-alone functions.
- Constants/variables can be defined globally, in classes, or methods.
- `bool` (vs. `boolean`)
- Explicit memory deallocation (*delete*)
- See:
  - [http://people.eecs.ku.edu/~miller/Courses/JavaToC++.html](http://people.eecs.ku.edu/~miller/Courses/JavaToC++.html)
  - Appendix A.12 in textbook
Examples

• See A1-basics.cpp
• See A1-fcopy.cpp
• See A1-CppJava -- .cpp / .java
• See A1-BookStoreCustomer -- .cpp / .java
Coding Conventions Used

• Need
  – to make it easier to read and maintain code
  – very important for large code bases
  – when multiple contributors

• Multiple coding styles are prevalent
  – people have differing tastes and preferences

• We impose some common coding practices for this class
Coding Conventions Used

• Make Files
  – to keep list of dependencies and to build all your project files
  – discussed further in Lab-1

• Header (.h) and implementation (.cpp) files
  – ADT interface (class definition) should be in .h file

• Comments
  – /* ... */ -- for block (multi-line) comments
  – // ... -- for single-line comments
Coding Conventions Used

• Indentation
  – 2 or 4 spaces
• For function definitions
  – open/close brace should be on its own line
  – block comment before each function tells what it does
• For braces within functions (loops/branches)
  – open brace should be on same line as construct
  – close brace should be on its own line
• Line width
  – a maximum of 80 characters on a single line
Coding Conventions Used

• Vertical white space and comments
  – blank line between consecutive code constructs
  – comments before important code constructs

• Horizontal white space
  – make it readable!

if((a==b) || (c<a))
over
if( ( a == b ) || ( c < a ) ) and
if((a==b) || (c<a))