SCIT-EIS-UOW
CSCI251/CSCI851 Advanced Programming
Autumn 2019

Laboratory Exercise 4 (Week 5)

1 Task One: Warm–up exercises

1. Debug: `Debug-A.cpp`. Relates to preprocessor directives. Note that logs and exponentials are inverses when you have a consistent base.
2. Some perspective: Look at `Scope.cpp`. Without compiling try to figure out which of the statements will work and which won't. For each of the comments add notes as to why or why not something will work.
3. This is a demo of some of the escaped output sequences, some of the time functionality of C++11, and the difference between `cout` and `cerr`.
   (a) Compile and run the program `Strange.cpp`.
   (b) Replace the `cerr` with `cout`, recompile and run. Explain what the difference is.

2 Task Two: A basic union

Look at `Union.cpp`.

1. Compile and run the program. What is being illustrated?
2. Change one of the types to a char. Compile and run.
3. Change the types so they have different sizes. Remember you can use `sizeof()` to find the size of a type.

3 Task Three: Pre–Processing

1. Add a debugging statement to one of your programs code using a DEBUG condition with a line number.
2. Write a program that uses PI defined from the command line compilation, accepts a radius from the user, then calculates:
   (a) The area of a circle with the entered radius.
   (b) The volume of a sphere with the entered radius.

4 Task Four: Programming defensively

1. Which of these implementations is better? Why?

```
size_T elements = strlen(container);
for (int i = 0; i < elements; ++i)
      state = combine(state, container[i]);

size_T elements = strlen(container);
for (int i = 0; i != elements; ++i)
      state = combine(state, container[i]);
```

2. Write a program that generates an SIZE by SIZE array containing random digits, where SIZE is defined during command line compilation. Use assert statements to ensure the value of SIZE is in the range 1 to 10. Note the problem with the SIZE will only be picked up in debug mode and only at run time.

5 Task Five: Follow the throwing

For the code in `Tracing.cpp` list, in order, the line numbers which are producing the output. Try and figure out where control is going prior to running the program.

6 Task Six: Some naming exercises

1. Set up an example illustrating how namespaces are discontiguous, even across multiple files.
2. Why is it generally bad practice to put the following in a header file.

```
using namespace some_namespace;
```

3. * How many layers of nesting are possible?
4. * How many layers of include are possible?