## Lab Experience 1

## Johnathan Lee

1. Delete // of the opening comment

**Error Message Generated:** “Missing type specifier“ and “Missing ; before ‘to’“.

**Type of Error:** Syntax error.

**What caused the error and how I corrected it:** The error was caused by the compiler trying to interpret the uncommented line. As the comment itself wasn’t a valid statement and didn’t contain any keywords, the compiler was unable to interpret it and spit out an error. The error was corrected by re-commenting out the line (Could have also simply deleted the comment text).

1. Comment out the preprocessor directive: #include <iostream>

**Error Message Generated:** “ ‘cout’: undeclared identifier “ And “ ‘endl’: undeclared identifier “.

**Type of Error:** Syntax error.

**What caused the error and how I corrected it:** The error was caused by the compiler being unable to find the previous definition/declaration of cout and endl in iostream. This can be fixed by either un-commenting the line again or making another include for iostream.

1. Put a semicolon at the end of the preprocessor directive

**Error Message Generated:** Warning: unexpected tokens following preprocessor directive. Expected a newline.

**Type of Error:** Syntax

**What caused the error and how I corrected it:** The warning was caused because the compiler found a semicolon where it shouldn’t be (after the include). Corrected by removing the semicolon.

1. Remove the right curly brace from the program.

**Error Message Generated:** “Syntax error: char should be preceded by ‘;’.” And “Missing type specifier – int assumed.”

**Type of Error:** Syntax.

**What caused the error and how I corrected it:** The error was caused because the compiler expected the main() to be either a full definition with both braces or a prototype (I think that’s the right term here) with a semicolon at the end. Corrected by re-adding the brace.

1. Replace the << with < in the first cout statement.

**Error Message Generated:** “binary ‘<’: no operator found which takes a left-hand operator of type ‘std::ostream’”

**Type of Error:** Syntax

**What caused the error and how I corrected it:** Error caused by a change from a stream operator to a comparison operator. The compiler was unable to find a way of comparing cout to things, so it ended compilation. Error fixed by changing the ‘<’ back to a ‘<<’.

1. Remove the apostrophes from the character constant ‘A’.

**Error Message Generated:** ‘A’: undeclared identifier.

**Type of Error:** Syntax.

**What caused the error and how I corrected it:** The compiler found A, which it saw as a symbol instead of the previous character. As it was unable to find a definition for A, it spit out an error. This could be corrected by either creating another variable named ‘A’ (and giving it a value) or re-adding the character marks (‘ ‘).

1. Change the apostrophes in ‘A’ to “”

**Error Message Generated:** ‘=’: cannot convert from ‘const char’ to char.

**Type of Error:** Syntax

**What caused the error and how I corrected it:** The variable letter was declared as a char, but changing the marks from single to double would indicate to the compiler that it should be treated as a full string. As it couldn’t find a way to convert between the two, it spat out an error. This could be fixed by either re-declaring letter as a string or changing the quotation marks back to singles.

1. Reverse the positions of the statements char letter; and letter = ‘A’;

**Error Message Generated:** ‘letter’: undeclared identifier.

**Type of Error:** syntax.

**What caused the error and how I corrected it:** Error was caused by the compiler not finding a previous definition of letter. As it had no definition of it, it didn’t know where to send the character being assigned to it (‘A’). Fixed by either moving the definition of letter to before it’s being used or merging the declaration and assignment.

1. Remove the \ from \n in the second cout statement.

**Error Message Generated:** No error.

**Type of Error:** Logic.

**What caused the error and how I corrected it:** Removing the escape character from the ‘n’s caused the compiler to interpret it as a normal ‘n’ instead of a newline. This could be corrected by either re-adding the escape character to the ‘n’s or adding “<< endl”s before and after the main message.

1. Place \ in front of both \n’s

**Error Message Generated:** None

**Type of Error:** Logic

**What caused the error and how I corrected it:** Opposite problem as above, adding in another escape character caused the compiler to escape the first escape character, meaning the ‘\n’ is now 2 separate characters. Once again, can be fixed by either removing the second ‘\’ or adding endl-s around the message.

1. Change the statement:

cout << "\nAnd the starry skies above...\n";

to

cout << "\nAnd the \”starry\” skies above...\n";

Did the output change? If yes, what was the change?

**Yes, it changed to: And the “starry” skies above…**

**Error Message Generated:** No error message.

**Type of Error:** Logic

**What caused the error and how I corrected it:** Not entirely an “error”, it caused the program to print out extra characters from what was originally expected.

1. Remove the statement return 0; from the program.

**Error Message Generated:** No message.

**Type of Error:** Technically syntax, but logic the way it worked for me.

**What caused the error and how I corrected it:** From memory, I’m 90% sure this should be giving me an error message about C++ not supporting default returns, but for some reason it’s refusing to give so much as a warning. Other than the error, I’d guess it could also be a logic error, as the program returns nothing to tell the OS what to do. Can be corrected by simply re-adding a return 0;