**Lab Experience Eight**

**Lab Exercises**

**Directions:**

Start Microsoft word and record the questions and answers to all of the exercises in your word document   
Answer the following questions based on material presented above and in chapters 1-6 of the textbook.

**Fill in the blank**

1. The word **void** precedes the name of the function prototype and heading indicating the function does not return a value.
2. Pass by **value** copies the contents of the actual parameter to the formal parameter.
3. Pass by **reference** passes the address of the actual parameter to the formal parameter.
4. When pass by value is used any changes made to the formal parameters contents will change the actual parameters contents. Is this statement True or is it False. **This statement is false.**
5. The for loop initialization expression is executed only **once**.

**Exercise 1**

You are interested in analyzing the frequency of numbers selected for the Powerball lottery. You have downloaded the previous year’s Powerball numbers which are saved in a text file called **pb\_2014.txt** and each record in the file has the following format:

01/01/14 15 24 40 48 52 23 x PP

01/04/14 19 20 37 41 58 14 x PP

01/08/14 10 28 39 47 58 22 x PP

01/11/14 10 15 33 48 54 34 x PP

Unfortunately this is an incompatible file format for the program you are using to analyze the frequency. The correct record file format needed by the program is shown below:

15 24 40 48 52 23

19 20 37 41 58 14

10 28 39 47 58 22

10 15 33 48 54 34

Write a program to convert the **pb\_2014.txt** file into the required file format shown above saving the new format to the file **powerball.txt**.

Copy and paste your program into your word document.

/\*

Programmer: Johnathan Lee

Lab#: 8

Description: Converts Powerball records into a usable format.

\*/

#include <fstream>

#include <iostream>

using namespace std;

int main() {

ifstream inFile;

ofstream outFile;

char line[18];

inFile.open("pb\_2014.txt");

if (!inFile) {

cout << "Error opening pb\_2014.txt for input. Terminating.";

exit(1106);

}

outFile.open("powerball.txt");

if (!outFile) {

cout << "Error opening powerball.txt for output. Terminating.";

exit(1106);

}

while(!inFile.eof()) {

inFile.ignore(10); // Ignore the date + 2 spaces.

inFile.get(line, 18); // Each record we need is exactly 17 characters.

outFile << line << endl;

inFile.ignore(6); // Each record we need is followed by 6 uneeded characters (x numNum + newline).

}

return 0;

}

**Exercise 2 (Line numbers are used for reference only.)**

1. #include <iostream>

using namespace std;

1. int multiply(int, int);
2. int main(){
3. int multiplier,multiplicand,product;
4. cout << "Enter the first number ===> " ;
5. cin >> multiplier;
6. cout << "Enter the second number ===> " ;
7. cin >> multiplicand;
8. product = multiply(multiplier, multiplicand);
9. cout << "Product of " << multiplier << " & " << multiplicand

<< " is " << product;

1. return 0;
2. }// end main
3. //
4. // Multiply two numbers and return their product
5. // Pre: num1 and num2 contain the values to be multiplied.
6. // Post: The product of num1 and num2.
7. int multiply(int num1, int num2){
8. return (num1 \* num2);
9. }

Download the above program from D2L and answer the questions below.

1. Comment out line 2 and describe what happens when you try to execute the program. Remove the comment after you have answered the question.

**Attempting to even build the program with line 2 commented out results in an identifier not found. Since the compiler goes top to bottom, it had not yet encountered the function multipier, and so didn’t know what the programmer was trying to reference.**

1. Change line 9 to the following: **multiply(multiplier, multiplicand);**

Describe what happens? If the assertion window appears, click on the ignore button. Change the statement back to the original once you have answered this question.

**Since the function does not take values by reference, the value ‘product’ doesn’t get assigned. Since it was never assigned beforehand, it still contains random garbage (in this case, -858993460).**

**Out of curiosity, could this be used for seeding random number generators, since whatever was previously in the memory in that location will already be fairly random?**

1. Change line 17 from **int multiply(int num1, int num2)** to **int multiply(int num1, num2)** . Describe what happens when you try to execute the program. Change the statement back to the original once you have answered the question.

**Since num2 no longer has a data type and the compiler reads char-by-char, it assumes that num2 is a data type. Since it is also unable to find any previous declaration for num2 that it could use for a data type, it throws an undeclared identifier.**

1. Comment out line 18 and describe what happens when you execute the program. Remove the comment after you have answered the question.

**Since multiply was declared as int and not void, it must always return a value. Commenting out line 18 removes the only possible return statement for the function, so the compiler throws an error at that point.**

1. Change line 2 to **void multiply(int, int );** and describe what happens when you execute the program. Change the statement back to the original once you have answered the question.

**Changing multiply’s return type to nothing causes anything that relied on a value it returned to not get data. Interestingly though, since void is apparently an actual data type instead of the lack of one, it actually seems to throw the error because there is no way to convert from void back to int. It also throws out errors because the definition of multiply says it returns an int, while the only possible prototype for multiply should return nothing.**

1. Change line 2 to **int multiply(int );** and describe what happens when you execute the program. Change the statement back to the original once you have answered the question.

**Since the earliest function prototype for multiply relied on only one input, the later function definition for multiply threw an error because it relied on two inputs.**

1. Change line 2 to **int multiply( );** and describe what happens when you execute the program. Change the statement back to the original once you have answered the question.

**Just like # 6, since the earliest prototype took a different number of arguments, the later function definition was, in the compiler’s mind, incorrect.**

1. Change line 9 to **product = int multiply(multiplier, multiplicand);** and describe what happens when you execute the program. Change the statement back to the original once you have answered the question.

**Since this is in the middle of an assignment statement, the compiler didn’t think it would find an ‘int’ here. Had the int or the following multiply() statement been surrounded by parentheses, it wouldn’t have thrown an error, though it would still be redundant (converting an int return to an int).**

1. Change line 9 to **product = multiply(multiplicand);** and describe what happens when you execute the program. Change the statement back to the original once you have answered the question.

**A little similar to #6 and #7, since the function prototype expects 2 inputs, attempting to call multiply with only one inputs results in an error because there is no matching function definition or prototype.**

1. Change line 18 to **return(multiplier \* multiplicand);** and describe what happens when you execute the program. Change the statement back to the original once you have answered the question.

**Since neither multiplier nor multiplicand were declared globally, using them outside the main function results in an undeclared variable. Although it would be very incorrect style-wise, the two variables could technically be declared globally, which would allow this approach to work.**

**Due Dates:**

As specified on the D2L assignment folder for Lab Experience Eight.

**What to hand in:**

1. Hand in a print out of your word document.
2. Hand in a print out of your program.
3. Compress the .cpp and the word processed document into a single compressed file called **{yourname}Lab8.zip** e.g. timwrennlab8.zip Note:**If your name is not part of the zip filename, I will not open the zipped file.**
4. Place the compressed file into the Lab Experience Eight assignment folder.