**ECE 15200: Programming for Engineers**

**Purdue University Northwest, ECE Department**

Laboratory 5: Function

**Instructions**:

* Submit only C++ source files (\*.cpp) for all the problems through Brightspace.
* Name each file following the format ***Lastname\_*Lab*X*\_p*Y*.cpp**, replace *Lastname, X,* and *Y* with your last name, lab #, and problem #, respectively.
* Put your name, assignment number, and date on the top of each source file (\*.cpp) as multi-line comment given below:

/\*

Class: ECE15200

Author: [Your Name]

Assignment: Lab [No.]

Date: [MM]/[DD]/[YY]

\*/

Remove the brackets after updating the information in them.

* PLEASE WORK ALONE. If any plagiarism is found, you will get ZERO. Never hesitate to discuss with the instructor/TA if stuck in any assignment problem.

**Problem 1.** (Lastname\_Lab5\_p1.cpp) Write a function named isPrime(int n) that accepts an integer n. If n is a prime number, the function returns 1, otherwise returns 0.

Write another function countPrime(int m) that accepts an integer number m. The function should return the number of the prime numbers between 2 and m, inclusive. Your countPrime function should call isPrime function and your main function should call countPrime function. [**20 points**]

**Problem 2.** (Lastname\_Lab5\_p2.cpp) Computers are playing an increasing role in education. Write a program that will help an elementary school student learn multiplication. Use rand() to produce two positive single digit integers. It should then display a question in the following format (assume 8 and 9 are randomly generated):

How much is 8 \* 9?

The student then types the answer. Your program checks the student’s answer. If it is correct, print “Very good!” If the answer is wrong, print “No. Please try again.” and then let the student try the same question again repeatedly until the student finally gets it right. If the number of incorrect exceeds five, the program should leave the question, display “Let’s try another one!” and ask a different question. After every answer, the program should ask whether the user wants to continue the program and based on the response, the program should be continued or terminated. [**20 points**]

**Problem 3.** (Lastname\_Lab5\_p3.cpp) Write a program that will conduct a quiz consists of 10 arithmetic questions and display the final score as ‘Your score in the quiz is *X*’ with a response message in the next line. A response message will be as follows based on the student’s performance:

1. Number of correct answers >= 9: Excellent, your are passed with ‘A’ grade!
2. Number of correct answers >= 7, but less than 9: Very Good, you are passed with ‘B’ grade!
3. Number of correct answers >= 5, but less than 7: Good, you are passed with ‘C’ grade!
4. Number of correct answers < 5: Sorry, we cannot pass you in the quiz!

For generating a question, you need to write a generateQuestion() function that will generate a question with the similar format as in **Problem 2**. In addition to generate the numbers randomly, the function should randomly select the operator among +, -, and \* operators. If the operator is + or -, the numbers should be double-digit. For \* operator, the numbers should be single-digit. The function should also compute the answer for the question to compare it with the student’s answer. If the answer is correct, the function returns 1, otherwise 0. Your main() function should call this function. [**20 points**]

**Problem 4**. (Lastname\_Lab5\_p4.cpp) Write a program that plays the game of “guess the number” as follows: Your program chooses the number to be guessed by selecting an integer at random in the range 1 to 100. The program then displays:

I have a number between 1 and 100.

Can you guess my number?

Please type your first guess:

The player then types a first guess. The program responds with one of the following:

1. Excellent! You guessed the number!

2. Too low. Try again.

3. Too high. Try again.

If the player’s guess is incorrect, your program should loop until the player finally gets the number right. Your program should keep telling the player “Too high” or “Too low” to help the player “zero in” on the correct answer. (Note: The search technique employed in this problem is called binary search in computer algorithm.) [**20 points**]

**Problem 5**. (Lastname\_Lab5\_p5.cpp) Write a function named minMax() that accepts three integer arguments from the keyboard and finds the smallest and largest integers. Include the function minMax() in a working program. Make sure your function is called from main(). Test the function by passing various combinations of three integers to it. [**20 points**]

**Problem 6**. (**Optional bonus**, Lastname\_Lab5\_p6.cpp) Write a function named sumFibonacci() that accepts a positive number n, find the value of , where indicates ith Fibonacci number. Include the function in a working program. Make sure your function is called from main(). Test the function by passing different values of n.

In mathematics, the Fibonacci numbers are the numbers in the following integer sequence, called the Fibonacci sequence, and characterized by the fact that every number after the first two is the sum of the two preceding ones: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, … . [**20 points**]