

CIS 230 – Spring 2006

Homework #3

Due Dates:

Part 1: Due in lab Monday, February 6, 2006 (To be shown to the instructor during lab. You will be allowed to keep these documents to finish your program)

- A description of the program (that you will include in your comments)
- Pseudocode for your program (see Psuedocode notes)

Part 2: Due by 10:00 AM, Wednesday, Feburary 8, 2006

- Your program code, your function header file, and your function definition file (at least 2 .cpp files and 1 .h file), also the function header file and definition file for the getRandomBetween function (see notes on this).
- Your script run (your typescript file – see “script” notes from homework #2)
- Use the submission tool on the qs-server to submit your files. You can access the tool by typing:
/class-files/gilden/230submit

Background Information:

The `rand` function generates "pseudo-random" numbers. That is, it produces a series of numbers that appear to be random. Every time it is invoked, it returns an "random" integer value between 0 and 32767. However, if you re-run your program, the same series of numbers will be produced. Once a program has been tested and you are sure it works, the random number function can be "seeded" with the `srand` function, so that it will return a different series of values each time the program is run. For this assignment, do *not* use `srand`!

An example:

```
#include <iostream >
using namespace std;

void main()
{
    for (int i = 0; i < 10; i++)
        cout<<rand()<<endl;
}
```

You should be able to predict what this program outputs. If you are not certain, type it in and execute it.

Sometimes we want a random number within a particular range, say for instance between 0 and 9. In this example, the "scaling factor" is 10. That is, there are ten digits inclusively between 0 and 9. If we use the modulus operator (%) to get the remainder of dividing the result of `rand` by 10, we will always get a value between 0 and 9. What will this program do?

```

#include <iostream>
using namespace std;

int main()
{
    for (int i = 0; i < 10; i++)
        cout<<rand()%10<<endl;

    return 0;
}

```

If we instead want a random number between 1 and 10, all we need to do is add 1 to the expression: `cout<<rand()%10 + 1`. What will this program do?

```

#include <iostream>
using namespace std;

int main()
{
    for (int i = 0; i < 10; i++)
        cout<<rand()%10+1<<endl;

    return 0;
}

```

In general, we can find a random value between two limits using the following algorithm:

```

scalingFactor ← highLimit - lowLimit + 1
temporaryResult ← rand() % scalingFactor
finalResult ← temporaryResult + lowLimit

```

What does this function do? How would you invoke it?

```

int getRandomBetween(int x, int y)
{
    int result;
    result = rand();
    result = result%(y - x + 1) + x;
    return result;
}

```

Program Description:

Write a program to help a second-grader practice his/her multiplication tables. Use the `rand` function to generate two positive numbers between 1 and 9. Then ask the student to provide the result of multiplying them together. Your program should prompt the student like this:

```

How much is 4 times 8?

```

The student then provides an answer. Your program should *not* accept a number less than 1! If the student's answer is wrong, print out "No, please try again.", repeat

the question, and continue until the student gets the answer right. When the student answers correctly, print out a message to help him/her remember: "Right, 4 times 8 is 32." If the student has answered correctly on the first try, your program should also print out this line: "Good job!" Once the student has answered correctly, ask the student if he or she would like to keep practicing. Your program should repeat until the student no longer wants to practice.

You *must* use a function to get the student's answer. Within that function, you must include code which does not allow the student to successfully enter a zero or a negative value. You may name your function as you like, but it should return an integer. The sole purpose of this function is to prompt the student for an answer, input that answer, and return it to the calling code - it should not evaluate whether the answer is correct. If a student enters a zero or a negative value, assume it's just a typing mistake. Such an answer doesn't count as "wrong".

You *must* declare your function in a .h file, using the format presented in class, and define it in a separate .cpp file.

You are free to use additional functions. If you do, they must be declared and defined in separate files.

Make sure that:

- a) your program does not accept a zero or negative number;
- b) your program forces the student to keep entering answers until s/he gets the answer correct (show at least two wrong responses and then the correct one);
- c) your program behaves as required when the student answers correctly on the first try.
- d) your program uses at least one function, as described above.
- e) your program uses two constants, to represent the "max" and "min" that can be used to get the random number

getRandomBetween

The function header and definition files (getRandomBetween.h and getRandomBetween.cpp respectively) have been provided for you on the qs-server. You can copy them to your current working directory by typing:

```
cp /class-files/gilden/hw03/getRandomBetween.* .
```

When you compile your program (assuming you have a main.cpp file), you can type:

```
g++ -c main.cpp
```

```
g++ -c getRandomBetween.cpp
```

```
g++ -c yourOtherFunction.cpp
```

```
g++ -o multiplyProgram main.o getRandomBetween.o yourOtherFunction.o
```

which will produce an executable file named "multiplyProgram". Feel free to name this as you wish. You can also type:

```
g++ -o multiplyProgram *.o
```