Assignment #2
Conditional & Repetitive Statements
Due: Friday, 01/31/14, 11:59pm

Grading: For each programming assignment, you are graded by explaining and demoing your code to a TA. **Your job is to convince the TA that your program works correctly, i.e. show your TA how to use/break your program☺** Your grade is based on Polya's 4 steps for solving problems:

- Understanding the problem. (*Recognizing what is asked*.) **(5 pts)**
- Devising a plan. (*Responding to what is asked*.) **(5 pts)**
- Carrying out the plan. (*Developing the result of the response*.) **(65 pts)**
- Looking back. (*Checking. What does the result tell me?*) **(5 pts)**

(65 pts) **Problem Statement:** Write a C++ program from your design in Assignment #1. Expand the problem from Assignment #1 to do the following:

- **Ask the user for a decimal (base 10) number, and display that number back to the user as a binary number.** You only need to deal with unsigned numbers, and you only want to print the significant bits that represent the base 10 number, i.e. 10 should be represented as 1010, not ...00000001010. In addition, you should allow the user to enter any number from 0 to the max positive number for the type you declare.
-
    - **How will you ensure that you get a number in the correct range**, i.e. if num is declared as an int, then you know the max is 2^(sizeof(int)*8-1)-1, right? What happens if the user enters a number larger than this? What happens if the user enter a number less than 0? **Your program must print an error message when the user enters an illegal input!!!**

- We learned from assignment #1 that **most numbers cannot be stored exactly in floating point notation** in the computer. If you know what to expect for an output, such as .01-.01, should be zero, then you can use a tolerance level to determine whether the loss of precision was acceptable. Because of this, there is a built-in tolerance for floats and doubles in the cfloat library, FLT_EPSILON and DBL_EPSILON. If you are within the tolerance level for that type, then you usually accept it as an acceptable loss of precision.

  You will get a float number and float epsilon value from the user. You will determine how many iterations of subtracting .01 it takes for the float to be outside of our tolerance level for precision. Do the same thing using a double and double epsilon value enter by the user. **Hint:** You want to use your knowledge that the more bits used for the floating point number, the more accurate the result.

(10 pts) **Program Style/Comments**
In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Below is an example header to include. Make sure you review the style guidelines for this class, and begin trying to follow them, i.e. don't align everything on the left or put everything on one line!
http://classes.engr.oregonstate.edu/eecs/winter2014/eecs161-001/161_style_guideline.pdf

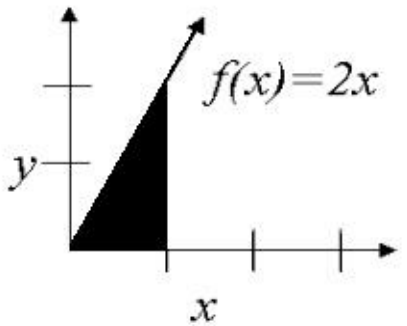
You are graded on having a header, proper comments, and readable code with indentation and vertical spacing that is CONSISTENT throughout your program. DO NOT align your entire program on the left side. This will cause you to automatically lose the full 10 points. In addition, do not forget your program header!!!

(10 pts) **Program Design for Assignment #3**

We are going to write a program in Assignment #3 to use the area of rectangles and/or trapezoids to find the area under a curve. We take the integral of a function to find the area under a curve, but how does a computer evaluate an integral? Is there an integrate function? Is there an integration symbol? Before you freak out, just remember that all we are doing is adding the area of rectangles or trapezoids together!!! ☺

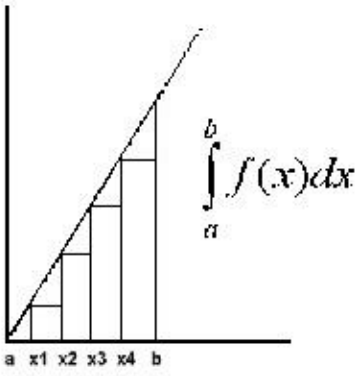
Objectives:
Analyze the different methods, rectangle vs. trapezoid methods, for solving integration problems. **Write a design and test cases** to calculate the area under a function using both the rectangle and trapezoid methods. Gain experience writing a short program, which uses selection, repetition, and functions/procedures.

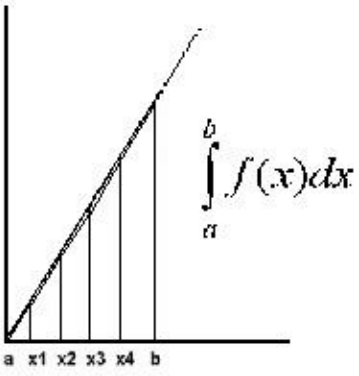
Background/Example:



$$\int_0^1 2x\,dx$$

Basic Numerical Methods (Rectangle vs. Trapezoid):





n = number of rectangles and/or trapezoids
a = beginning x value
b = ending x value
w = width of each rectangle/trapezoid, (b-a)/n
f(x) = height of rectangle
(f(x1)+f(x2))/2 = height of trapezoid
Area = width*height

Integration Program Definition:

Your program should determine the area under these specific functions,

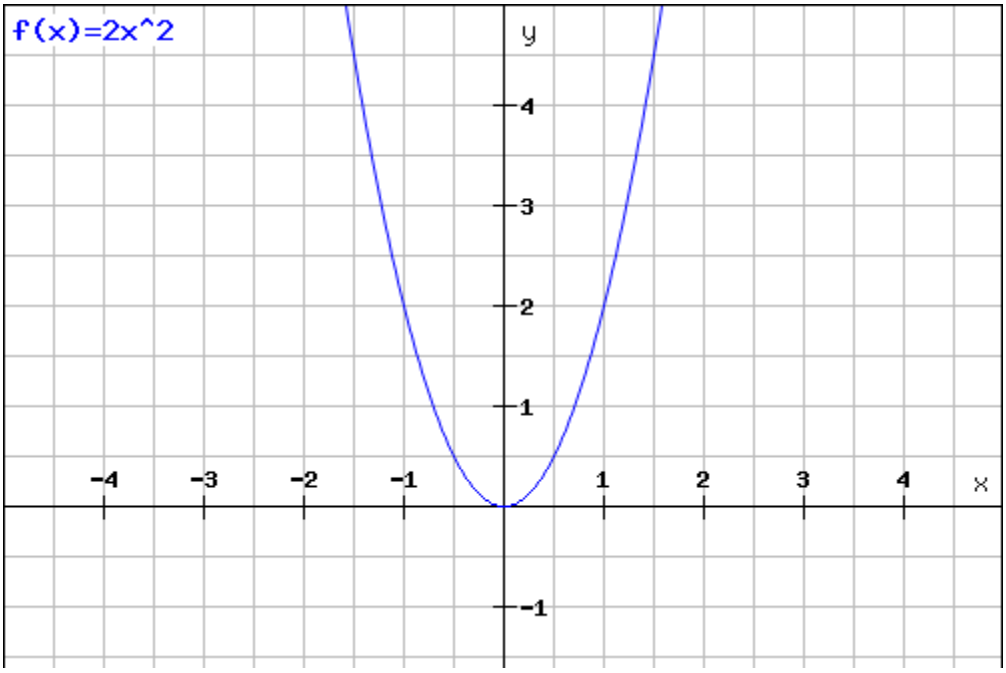
f1(x) = $2x^5 + x^3 - 10x + 2$
f2(x) = $6x^2 - x + 10$
f3(x) = $5x + 3$
f4(x) = $2x^3 + 120$
f5(x) = $2x^2$

The functions are bounded by any interval on the x-axis, including both positive and negative values!!!. You can check your answers to these functions using http://calc.begin-programming.com/

The area calculated is determined by the **user's choice of function and method** to use to calculate the integral, and it is possible for the user to choose to see the area for one function calculated by both methods. **For example:**

Choose a function (1, 2, 3, 4, 5, other(quit)): **1**
Would you like to calculate the area using the rectangle, trapezoid, or both (1, 2, 3): **2**
How many trapezoids do you want? **1000**
Please select a starting point, a = **1**
Please select an ending point, b = **2**
The area under **$2x^2$** between 1 and 2 is 4.667

Plot of **f5(x)= $2x^2$**



Program Description in more detail:
Your program needs to adhere to the following guidelines:

- **If the user chooses to see the area calculated by both methods, each method should receive their own number of rectangles or trapezoids as input and return the value from the calculation.**
- **Your program should continue running until the user no longer wants to calculate the area under a curve.**
- **You should use procedural decomposition and have functions for f1(x), f2(x), f3(x), f4(x), and f5(x), as well as functions for calculating the area using the rectangle vs. trapezoid method.**

Program Input:
- Starting and ending points for the area
- Function to calculate the area, i.e. f1, f2, f3, f4, f5
- Function/Procedure(s) for calculating the area, i.e. rect, trap, both
- Number of rectangles and/or trapezoids to use

Program Output:
- The function being evaluated
- Starting and ending points for the integral
- Number of rectangles and/or trapezoids used
- The area calculated by the method(s)

Electronically submit your C++ program (**.cpp** file, not your executable!!!) and design document, **as a pdf**, by the assignment due date, using **TEACH**.