Brandon Lee

EECS 161 R1000

Assignment 2

31 January 2014

1. **Understanding the Problem**
   1. The first part of the problem is asking for me to convert a user inputted number into its binary form. The second part of the assignment is asking for the user to input a float number and float epsilon value and outputs the number of iterations of subtracting 0.01 for the float to become under the accepted tolerance. Same for doubles.
2. **Devising a Plan/Design**
   1. Binary Converter
      1. define variables for problem
      2. utilize void to define binary function
   2. Tolerance functions
      1. Define user inputted float/double and the user inputted epsilon values for both as well
      2. Define a variable that stores the number of iterations as well
      3. Prompt user to type in user inputted variables
      4. Utilize for loop to subtract 0.01 from the user inputted float/double
      5. Have loop add 1 to iteration variable for each time it loops
      6. Break when float/double becomes under its respective epsilon
3. **Looking Back/Self-Reflection**
   1. Looking back, I find that I could have made the code much more readable for myself during my initial attempts on the assignment. Including a couple more comments would have been nice as well.
   2. From this assignment, I learned how to define and call void functions. I also learned how to utilize while loops in my code.
4. **Design for Assignment #3**
   1. Since integration is essentially finding the area under the curve of a function, we can accomplish this by dividing up the area underneath the curve into either rectangles or trapezoids. The function will first allow the user to input an equation to integrate. Next, the function will ask if the user would like to integrate using either rectangles or trapezoids. If the user prompts for rectangles, we will implicate a Riemann sum technique and find the “n”, “delta x”, “b”, and “a” of the equation. If the user prompts for the trapezoid, we will implicate the trapezoidal rule.