Student Name: Paula Klotz

Date: February 9, 2017

Assignment: Week 4 – Testing

**Average:**

* Sample input values: 2, 2, 2 (test 1); 0,0,0 (test 2); 2,2,1 (test 3); -2,-2,-2 (test 4) . The correct output should be 2 (test 1), 0 (test 2), 1.67 (test 3), -2 (test 4).
* After implementing the function Test\_Average(), the expected output did not match my sample output listed above. The test failed.
* Test\_Average() code:

void Test\_Average()

{

//Test 1

if (Average(2, 2, 2) != 2)

{

cout << "Test 1 Failed 2,2,2" << endl;

}

else

{

cout << "Test 1 Passed 2,2,2" << endl;

}

//Test 2

if (Average(0, 0, 0) != 0)

{

cout << "Test 1 Failed 0,0,0" << endl;

}

else

{

cout << "Test 1 Passed 0,0,0" << endl;

}

//Test 3

if (Average(2, 2, 1) != 1.67)

{

cout << "Test 1 Failed 2,2,1" << endl;

}

else

{

cout << "Test 1 Passed 2,2,1" << endl;

}

//Test 4

if (Average(-2, -2, -2) != -2)

{

cout << "Test 1 Failed -2,-2,-2" << endl;

}

else

{

cout << "Test 1 Passed -2,-2,-2" << endl;

}

}

* Average (float,float,float) function revised to read:

float Average(float a, float b, float c)

{

return (a + b + c) / 3;

}

* **What was wrong with the float function?** The calculation was missing parenthesis around the three parameters (a,b,c).

**Max:**

* Sample input values: 1, 2, 3. The correct output should be 3.
* After implementing the function Test\_Max(), the expected output did not match my sample output listed above. The test failed.
* Test\_Max() code:

void Test\_Max()

{

//Test 1

if (Max(1, 2, 3) != 3)

{

cout << "Test 2 Failed 1,2,3" << endl;

}

else

{

cout << "Test 2 Passed 1,2,3" << endl;

}

//Test 2

if (Max(2, 2, 3) != 3)

{

cout << "Test 2 Failed 2,2,3" << endl;

}

else

{

cout << "Test 2 Passed 2,2,3" << endl;

}

//Test 3

if (Max(3, 2, 1) != 3)

{

cout << "Test 2 Failed 3,2,1" << endl;

}

else

{

cout << "Test 2 Passed 3,2,1" << endl;

}

//Test 4

if (Max(-1, 6, 0) != 6)

{

cout << "Test 2 Failed -1,6,0" << endl;

}

else

{

cout << "Test 2 Passed -1,6,0" << endl;

}

}

* Max(int,int,int) function revised to read:

int Max(int a, int b, int c)

{

if (a > b && a > c)

{

return a;

}

else if (b > a && b > c)

{

return b;

}

else

{

return c;

}

}

* **What was wrong with the max function?**  I added relational operators to the first if statement to read if (a > b && a >), and changed the return result to ‘a’. I added an else if statement with relational operators to read else if (b > a && b > c), and changed the return result to ‘b’. Then the third return result was changed to ‘c’.

**Factorial:**

* Sample input values: 4. The correct output should be 24.
* After implementing the function Test\_Factorial(), the expected output did not match my sample output listed above. The test failed.
* Test\_Factorial() code:

void Test\_Factorial()

{

//Test 1

if (Factorial(4) != 24)

{

cout << "Test 3 Failed 4" << endl;

}

else

{

cout << "Test 3 Passed 4" << endl;

}

//Test 2

if (Factorial(-4) != -24)

{

cout << "Test 3 Failed -4" << endl;

}

else

{

cout << "Test 3 Failed -4" << endl;

}

}

* Factorial(int) function revised to read:

int Factorial(int n)

{

int fac = 1;

for (int i = n; i > 0; i--)

{

fac = fac \* i;

}

return fac;

}

* **What was wrong with the factorial function?** The int fac should have been initialized to 1 versus the parameter ‘n’. Within the for statement, I changed the Boolean expression to read i > 0 versus i >= 0.

**FormatName:**

* Sample input values: Paula, Klotz. The correct output should be Klotz,Paula.
* After implementing the function Test\_FormatName(), the expected output did match my sample output listed above. The test passed.
* Test\_FormatName(string,string) code:

void Test\_FormatName()

{

if (FormatName("Paula", "Klotz") !="Klotz,Paula")

{

cout << "Test 4 Failed" << endl;

}

else

{

cout << "Test 4 Passed" << endl;

}

}

* FormatName (string,string) function revised to read:

string Format\_Name(string first, string last)

{

return last + "," + first;

}

* **What was wrong with the FormatName function?** I added an underscore to the function name to read Format\_Name. Everything else is okay.