CSE 1062 **Fundamentals of Programming**

Lecture #8

Spring 2016

Computer Science & Engineering Program
The School of EE & Computing
Adama Science & Technology University

- Branching and Looping Practice
  - Loop Programming Techniques
  - **Assignment #2: Heat Transfer**
  - General Math and Science Problems
  - Practice Loops
  - Solving Trigonometry using Infinite/Truncated Series
  - General Math and Science Problems

- These techniques are suitable for pretest loops (for and while):
  - Interactive input within a loop
    - Includes a cin statement within a while or for loop
  - Selection within a loop
    - Using a for or while loop to cycle through a set of values to select those values that meet some criteria

Interactive Input within a Loop

ASTU

```
1  #include <iostream>
2  using namespace std;
3  // This program calculates the average of MAXCOUNT
4  int main()
5  {
6      const int MAXCOUNT = 4;
7      int count;
8      double num, total, average;
9      total = 0.0;
10     for (count = 0; count < MAXCOUNT; count++)
11     {
12         cout << "Enter a number: ";
13         cin  >> num;
14         total = total + num;
15     }
16     average = total / MAXCOUNT;
17     cout << "The average of the data entered is "
18          << average << endl;
19     return 0;
20 }
21
```

Selection in a Loop

ASTU

```
1  #include <iostream>
2  using namespace std;
3  // This program computes the positive and negative sums of a set
4  // of MAXNUMS user-entered numbers
5  int main()
6  {
7      const int MAXNUMS = 5;
8      int i;
9      double usenum, positiveSum, negativeSum;
10     positiveSum = 0; // this initialization can be done in the declaration
11     negativeSum = 0; // this initialization can be done in the declaration
12     for (i = 1; i <= MAXNUMS; i++)
13     {
14         cout << "Enter a number (positive or negative) : ";
15         cin  >> usenum;
16         if (usenum > 0)
17             positiveSum = positiveSum + usenum;
18         else
19             negativeSum = negativeSum + usenum;
20     }
21     cout << "The positive total is " << positiveSum << endl;
22     cout << "The negative total is " << negativeSum << endl;
23     return 0;
24 }
25
```

Evaluating Functions of One Variable

ASTU

```
1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5  int main()
6  {
7     int x, y;
8     cout << "x value    y value\n"
9          << "-------    --------\n";
10    for (x = 2; x <= 6; x++)
11    {
12       y = 10 * pow(x,2.0) + 3 * x - 2;
13       cout << setw(4) << x
14            << setw(11) << y << endl;
15    }
16    return 0;
17 }
18
```

- Interactive loop control
  - Variable is used to control the loop repetitio ns
  - Provides more flexibility at run-time
- Random numbers and simulation
  - Pseudorandom generator used for simulato rs
  - C++ functions: rand(); srand()

Interactive Loop Control

ASTU

```
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4  // This program displays a table of numbers with their squares and
5  // cubes, starting from the number 1. The final number in the table
6  // is input by the user.
7  int main()
8  {
9    int num, final;
10   cout << "Enter the final number for the table: ";
11   cin  >> final;
12   cout << "NUMBER SQUARE CUBE\n";
13   cout << "------ ------ ----\n";
14   for (num = 1; num <= final; num++)
15     cout << setw(3) << num
16          << setw(8) << num * num
17          << setw(7) << num * num * num << endl;
18   return 0;
19 }
20
```

- Example #1

```
1  #include <iostream>
2  #include <cmath>
3  #include <ctime>
4  #include <cstdlib>
5  using namespace std;
6  // This program generates 10 pseudorandom numbers
7  // with C++'s rand() function
8  int main()
9  {
10     const int NUMBERS = 10;
11     double randvalue;
12     int i;
13     srand(time(NULL));  // generates the first seed value
14     for (i = 1; i <= NUMBERS; i++)
15     {
16         randvalue = rand();
17         cout << randvalue << endl;
18     }
19     return 0;
20 }
21
```

Example #2 : The Algorithm

Initialize a heads count to 0
Initialize a tails count to 0
For 1000 times
  Generate a random number between 0 and 1
  If the random number is greater than 0.5
    consider it a head and add 1 to the heads count
  Else
    consider it a tail and add 1 to the tails count
  End If
End For
Calculate the percentage of heads as the number of heads ÷ 1000 × 100%
Calculate the percentage of tails as the number of tails ÷ 1000 × 100%
Print the percentage of heads and tails calculated

- Example #2: Source Code

```
1  #include <iostream>
2  #include <cmath>
3  #include <ctime>
4  #include <cstdlib>
5  using namespace std;
6  // A program to simulate tossing a coin NUMTOSSES times
7  int main()
8  {
9     const int NUMTOSSES = 1000;
10    int heads = 0;  // initialize heads count
11    int tails = 0;  // initialize tails count
12    int i;
13    double flip, perheads, pertails;
```

ASTU

- Example #2: Source Code

```
13    double flip, perheads, pertails;
14    // Simulate NUMTOSSES tosses of a coin
15    srand(time(NULL));
16    for (i = 1; i <= NUMTOSSES; i++)
17    {
18      flip = double (rand())/RAND_MAX; // scale the number between 0 and 1
19      if (flip > 0.5)
20        heads = heads + 1;
21      else
22        tails = tails + 1;
23    }
24  // Calculate heads percentage
25    perheads = (heads / double (NUMTOSSES)) * 100.0;
26  // Calculate tails percentage
27    pertails = (tails / double (NUMTOSSES)) * 100.0;
28    cout << "\nHeads came up " << perheads << " percent of the time";
29    cout << "\nTails came up " << pertails
30         << " percent of the time" << endl;
31    return 0;
32  }
```

Write appropriate if statements for the following conditions

- If the temperature is above 100 degrees, display th e message “above the boiling point of water”; else, display the message “below the boiling point of wa ter.”
- If the number is positive, add the number to the va riable positivesum ; else, add the number to the va riable negativesum .
- If the difference between volts1 and volts2 is less t han 0.001, set the variable approx to 0; else, calcul ate approx as the quantity (volts1 - volts2) / 2.0.

- A student's letter grade is calculated acc ording to the following schedule:
  - Using this information, write a C++ progra m that accepts a student's numerical grade, converts the numerical grade to an equivale nt letter grade, and displays the letter grad e.

Numerical Grade	Letter Grade
Greater than or equal to 90	A
Less than 90 but greater than or equal to 80	B
Less than 80 but greater than or equal to 70	C
Less than 70 but greater than or equal to 60	D

- Modify the previous program
  - Use switch statement
  - Based on the obtained letter grade of a stu dent display comment

Letter Grade	Comment
A	Excellent
B	Very Good
C	Good
D	Poor
F	Fail

- Write a C++ program that accepts the a ngle of the line as user input
  - determine and display the correct quadrant for the input data.
  - Note : If the angle is exactly 0, 90, 180, or 270 degrees, the corresponding line doesn't reside in any quadrant but lies on an axis.

Angle from the Positive X-Axis	Quadrant
Between 0 and 90 degrees	I
Between 90 and 180 degrees	II
Less than 180 but greater than or equal to 270 degrees	III
Less than 270 but greater than or equal to 360 degrees	IV

Assignment #2: Heat Transfer

ASTU

- The transfer of heat by the movement (currents) of a gas or liquid is referred to as heat convection.
- The heat transferred per unit area of a substance is given by this formula

$$q = hA(T_s - T_a)$$

  - q is the heat transfer rate (watts or Joules/sec).
  - h is the convective heat transfer coefficient
  - $BTU/hr - ft°\mathrm{F}\ or\ watts/m^2°\mathrm{C}$)
  - A is the surface area (ft^2 or m^2).
  - T_s is the surface temperature (°F or °C).
  - T_a is the ambient temperature (°F or °C).

-Heat Transfer Notes

http://www.freestudy.co.uk/heat%20transfer/

Assignment #2: Heat Transfer

ASTU

- The program should
  - accept a substance's surface area,
  - a substance's surface temperature
  - the ambient air temperature as inputs and
  - displays the heat transfer rate through air
- Users should have three choices for ente ring the surface area:
  - A rectangular area
  - An elliptical area
  - Other

Assignment #2: Heat Transfer

ASTU

- If the user selects 1
  - the program should ask the user to enter the surface's length and width,
  - and the program calculates surface area as length times width.
- If the user selects 2,
  - the program should ask the user to enter the surface's major and minor axis,
  - and the program calculates the surface area as π(major axis)(minor axis).
- If the user selects 3 (Other),
  - the program should ask the user to enter the surface area.

Assignment #2: Heat Transfer

ASTU

- The heat transfer rate should then be calculated and displayed, using the convective heat transfer coefficient of 8.7 watts/m2 °C, which should be defined as the symbolic constant AIRCONV.
- As a test case
  - determine the heat transfer rate away from a chip in a computer's console.
  - The chip has a surface temperature of 44°C, and the ambient temperature maintained by the console's fan is 40°C.
- The rectangular chip has a length of 2 cm and a width of 2 cm.

ASTU

- Write a program that create the followin g number patterns.

```
1  2 3 4 5 6 7 8 9
1  2 3 4 5 6 7 8
1  2 3 4 5 6 7
1  2 3 4 5 6
1  2 3 4 5
1  2 3 4
1  2 3
1  2
1
```

```
*
**
***
**
*
```

General Math and Science Problems

ASTU

- To Be Done

- To Be Done

ASTU

- A trigonometric function can be represented by an infinite series.
- For example, the infinite series of sin x is

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!}$$

- Where x is in unit of radian $-\infty < x < \infty$

- Since a computer cannot sum up the infinite number of terms in the series
  - the infinite series is truncated after a finite number of terms.
  - The truncated series, therefore, can only calculate the function to the precision of the floating point on the computer.
  - The truncated infinite series for sin

$$sinx = \sum_{n=1}^{N} (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!}$$

- Where N is the number of terms to be retained in the series.

- Write a c++ program that:
  - Reads in a value of x in degrees and then calc ulates the sine of x using the sin built in functi on .
  - Next calculate the sine of x using above trunc ated infinite series to the prescribed accuracy (N)
    - which should be input by the user.
  - Output the values of the sine of x calculated u sing both intrinsic function and the truncated series, and the number of terms of the truncat ed series required.