Laboratory VI – Program Control Using Loops

Larry Caretto
Computer Science 106

Computing in Engineering and Science

March 9, 2006

California State University Northridge

Outline

- Exercise six goals
- Outline tasks for exercise six
- Introduce idea of nested loops and table generation
- Provide details for some tasks

Exercise Six Goals

- As a result of this exercise you should be able to accomplish the following:
  - write looping structures using both the while and for commands
  - write programs with nested loops
  - prepare a table of values for a function of two variables
  - write a program to compute the sum of an infinite series

Tasks for Exercise Six

- One – copy and paste code to produce table of kinetic energy as a function of mass and velocity
- Two – modify task one code to create similar table for a different formula
- Three – write code with a while loop to sum an infinite series for sin(x)
- Four – modify task three code to use a for loop in place of a while loop

Task One: Kinetic Energy Table

- $KE = mV^2/2$
- Copy and execute code from exercise
- Code prints table of KE as a function of mass, m, and velocity, V
  - $1\ kg \le m \le 25\ kg$, with $\Delta m = 1\ kg$
  - $6\ m/s \le V \le 15\ m/s$, with $\Delta V = 1\ m/s$
- Modify this code for task two

Nested `for` Loops

- Can have an inner `for` loop nested inside an outer `for` loop
- Example, print table of kinetic energy such as the one below

```
   Velocity values m/s below
     6    7     8     9    10
Mass
1  18.0  24.5  32.0  40.5  50.0
2  36.0  49.0  64.0  81.0 100.0
```

Kinetic Energy Table

- Use nested for loops
- Inner loop calculates and prints one row of the table
- Outer loop does inner loop for all rows
- Use type int variables for loop indices
- Convert to double before division by 2
- Need initial loop to print headers for each column

One Table Row

- Each row prints a mass then prints the KE for each velocity from 6 to 15
- What is loop index for printing a row?

```
cout << setw(3) << m;
for( int v = 6;  // initialize
       v <= 15;  // continue
       v++ )     // increment
{
```


Full Code for One Table Row

```
cout << setw(3) << m;
for( int v = 1; v <= 10; v++ )
{
    double KE = double( m * v
                * v ) / 2;
    cout << setw(7) << KE;
}
```

from previous chart

How to Get Table?

- Print column header row
- Loop over all values of mass
  - Move output to a new line
  - For each value of mass, use code just developed to print one row
- End loop over mass

Code to Produce Table

```
// Put column header code here
for( int m = 1; m <= 10; m++ )
{
    cout << endl;

 //Code for one row
}
cout << endl;
```


Task One Code

```
for ( int V = 6; V <= 15; V++ )
{     // print column header with spacing
}
for ( int m = 1; m <= 25; m++ ) //row loop
{
     cout << "\nm = " << setw(2) << m;
     for ( int V = 6; V <= 15; V++ ) //cols
     {
        double KE = 0.5 * m * V * V;
        cout << setw(7) << KE;
      }
}
```

Task Two: Table of A/P Ratio

$\frac{A}{P} = \frac{i}{1-(1+i)^{-n}}$	$= \frac{0.015}{1-(1+0.015)^{-10}} = 0.10843$
Formula	Example for i = 1.5%, n = 10

- Modify task one code to prepare table of recurring payment ratio, A/P
- Function of interest rate, i, and periods, n
  - 0.05% ≤ i ≤ 2%, with Δi = 0.05%
  - 6 ≤ n ≤ 36, with Δn = 6

California State University Northridge 13

Loop termination problems

- Numbers are not represented exactly in the computer
- Code like the following may not give correct end point due to roundoff error

```
for ( double i = .01; i <= .1; i += .005)
```

- Suggested alternatives

```
for ( double i = .01; i <= .102; i += .005)
for ( int count = 0;  count <= 18; count++)
{
    double i = 0.01 + 0.005 * count;
```

California State University Northridge Could have i = 0.100000000000001 14

Other Task Two Issues

- Spacing for output
- Which is in rows and which is in columns? (Hint: what will fit?)
- Getting column headers to line up with numbers in columns
- Note initial spacing in column header provided in initial cout statement as blank string, “ “

California State University Northridge 15

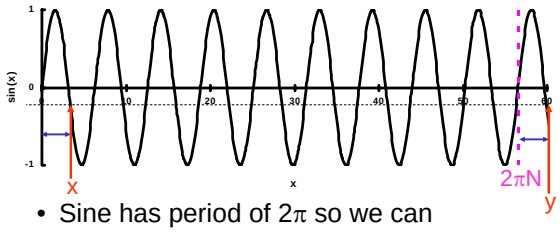
```
cout << "                  Table of Kinetic
     << "       Masses, m,  in kilograms and
     << "V, in meters per second.\n\n"
     << "          " << fixed << setprecision(1);
for ( int velocity = 6; velocity <=15;
                              velocity++ )
{
   if ( velocity < 10 )  // adjust format
   {
      cout << "  V = " << velocity;
   }        // 2 spaces before V =
   else
   {
      cout << " V = " << velocity; V
   }        // 1 space before V=
}
```

Tasks Three and Four

- Write code to evaluate infinite series for sine of an angle, x
- Do this using both a for loop and a while loop
- For large angles, compute the sine of an equivalent angle between 0 and 2π
- See exercise for more details on computation of infinite series for e^x and modification of this code to compute sin

California State University Northridge 17

Equivalent Angles



- Sine has period of 2π so we can compute the sine of a large angle, y, as sine of x = y – 2πN, where N = int(x/2π)

California State University Northridge 18

Infinite Series for e^x

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \qquad n! = n(n-1)(n-2)\cdots(3)(2)(1)$$
$$(n-1)! = (n-1)(n-2)\cdots(3)(2)(1)$$

$$n! = n(n-1)! \qquad or \qquad (n-1)! = \frac{n!}{n}$$

$$(n-1)! = \frac{n!}{n} \Rightarrow 1! = (2-1)! = \frac{2!}{2} = 1 \quad and \quad 0! = (1-1)! = \frac{1!}{1} = 1$$

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = \frac{1}{0!} + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \cdots$$

California State University Northridge 19

Ratio of Terms in e^x Series

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = \sum_{n=0}^{\infty} T_n \quad \Rightarrow \quad T_n = \frac{x^n}{n!}$$

$$\frac{T_n}{T_{n-1}} = \frac{\frac{x^n}{n!}}{\frac{x^{n-1}}{(n-1)!}} = \frac{x^n}{x^{n-1}} \frac{(n-1)!}{n!} = x \frac{(n-1)!}{n(n-1)!} = \frac{x}{n}$$

California State University Northridge 20

Coding the Series

$$e^x = \sum_{n=0}^{\infty} T_n \quad \text{where} \quad T_0 = 1 \quad and \quad T_n = \frac{x}{n} T_{n-1}$$

- Code for this approach

```
newTerm = oldTerm * x / n;
seriesSum = seriesSum + newTerm
oldTerm = newTerm;
```

California State University Northridge 21

Alternative Coding for Series

- Code from last page

```
newTerm = oldTerm * x / n;
seriesSum = seriesSum + newTerm
oldTerm = newTerm;
```

- Simpler Code uses one term variable

```
term = term * x / n;
sum = sum + term;
```

- Still Simpler Code

```
term *= x / n;
sum += term;
```

- Must initialize sum and term properly

California State University Northridge 22

Starting the Loop

```
const int maxN = 100;
const double maxError = 1e-12;
bool converged = false;
double term = 1; // change this!
double sum = term;
int n = 0;
while ( !converged && n < maxN)
{
     // see next slide
```

California State University Northridge 23

Loop Body

```
while ( !converged && n < maxN)
{
    n++;
    term *= x / n;
    sum += term;
    converged = fabs( term )
         <= maxError * fabs( sum );
}
```

California State University Northridge 24

Why did we exit the loop?

```
if ( converged )
{
    cout << "For x = " << x <<
         ", exp(x) = " << sum;
}
else
{
    cout << "No Convergence";
}
```


Sine Series

$$\sin(x) = \sum_{n=0}^{\infty} T_n = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots$$

$$\frac{T_n}{T_{n-1}} = \frac{\frac{(-1)^n x^{2n+1}}{(2n+1)!}}{\frac{(-1)^{n-1} x^{2(n-1)+1}}{[2(n-1)+1]!}} = \frac{(-1)^n}{(-1)^{n-1}} \frac{x^{2n+1}}{x^{2n-1}} \frac{(2n-1)!}{(2n+1)!} = \frac{-x^2}{2n(2n+1)}$$

- Similar, but more complicated, than e^x
- First term in series is x, not 1

Using a `for` Loop

- Continuation condition can be complex
- Remember while condition for series
  `while ( !converged && n < maxN)`
- Can have similar condition in for loop
- Can also have multiple initializations or conditions separated by a comma

```
for ( n = 0, converged = false;
  !converged && n < maxN; n++ )
```

- Watch first n value and increment

Style: Indent Structures

```
while ( inFile.good() )
{
    inFile << hours << rate;
    if ( hours > 40 )
        pay = rate * ( 40 +
           1.5 * ( hours – 40 ) );
    else
        pay = rate * hours;

    outFile << pay << endl;
}
```


Bad Style

```
while(inFile.good()){inFile<<hours<<
rate;if(hours>40)pay=rate*(40+1.5*(
hours-40));else pay=rate*hours;
outFile<<pay<<endl;}
```