`switch` Statement Example

```
int alpha = 0; int beta = 0; int gamma = 0;  int n;
cout << "Please enter a value of n for processing " << endl;
cin >> n;
cout << " The value you entered for n is " << n << endl;

switch(n)
{
   case 3:  alpha++;
            break;
   case 7:  beta++;
            break;
   case 10: gamma++;
            break;
}
cout << "alpha is " << alpha << endl << "beta is " << beta
     << endl  << "gamma is " << gamma << endl;
}
```


Loop Example

```
#include <iostream>
using namespace std;

int main()
{
   int sum;
   int count;

   sum = 0;
   count = 1;
   while (count <= 1000)
   {
      sum = sum + count;
      count++;
   }
   cout << " The final sum is " << sum << endl;
}
```


Equivalent `for` Statement

```
#include <iostream>
using namespace std;

int main()
{
   int sum;
   int count;

   sum = 0;
   for (count = 1; count <= 1000; count++)
      sum = sum + count;
   cout << " The final sum is " << sum << endl;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                  i = 4
   {
      for (j = 0; j < i; j++)
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                  i = 4
   {
      for (j = 0; j < i; j++)                j = 0
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                  i = 4
   {
      for (j = 0; j < i; j++)                j = 0
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```

UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 4
   {
      for (j = 0; j < i; j++)              j = 1
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 4
   {
      for (j = 0; j < i; j++)              j = 1
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 4
   {
      for (j = 0; j < i; j++)              j = 2
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 4
   {
      for (j = 0; j < i; j++)              j = 2
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 4
   {
      for (j = 0; j < i; j++)              j = 3
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 4
   {
      for (j = 0; j < i; j++)              j = 3
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```

Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 4
   {
      for (j = 0; j < i; j++)              j = 4
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 4
   {
      for (j = 0; j < i; j++)              j = 4
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 3
   {
      for (j = 0; j < i; j++)              j = 4
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 3
   {
      for (j = 0; j < i; j++)              j = 0
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 3
   {
      for (j = 0; j < i; j++)              j = 0
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 3
   {
      for (j = 0; j < i; j++)              j = 1
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```

Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 3
   {
      for (j = 0; j < i; j++)              j = 1
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 3
   {
      for (j = 0; j < i; j++)              j = 2
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 3
   {
      for (j = 0; j < i; j++)              j = 2
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 3
   {
      for (j = 0; j < i; j++)              j = 3
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 3
   {
      for (j = 0; j < i; j++)              j = 3
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 2
   {
      for (j = 0; j < i; j++)              j = 3
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```

Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 2
   {
      for (j = 0; j < i; j++)              j = 0
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 2
   {
      for (j = 0; j < i; j++)              j = 0
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 2
   {
      for (j = 0; j < i; j++)              j = 1
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 2
   {
      for (j = 0; j < i; j++)              j = 1
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 2
   {
      for (j = 0; j < i; j++)              j = 2
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 2
   {
      for (j = 0; j < i; j++)              j = 2
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```

UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 1
   {
      for (j = 0; j < i; j++)              j = 2
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 1
   {
      for (j = 0; j < i; j++)              j = 0
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 1
   {
      for (j = 0; j < i; j++)              j = 0
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 1
   {
      for (j = 0; j < i; j++)              j = 1
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 1
   {
      for (j = 0; j < i; j++)              j = 1
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


UAH CPE 112

Nested for Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                i = 0
   {
      for (j = 0; j < i; j++)              j = 1
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```

Nested `for` Example

```
#include <iostream>
using namespace std;

int main()
{
   int i;
   int j;

   for (i = 4; i >= 1; i--)                  i = 0
   {
      for (j = 0; j < i; j++)                j = 1
         cout << j << "****";
      cout << i << endl;
   }
   return 0;
}
```


`do-while` Statement Example

```
int main()
{
   int num; int count = 0; int sum = 0;

   do
   {
      cout << "Please enter a number for summing " << endl;
      cin >> num;
      if (num > 0)
      {
         sum = sum + num;
         count++;
         cout << "The number is " << num << endl;
      }
   } while ((num > 0) && (count < 10));
   cout << "The sum is " << sum << endl;
}
```


Problem Solving Case Study

- Problem: Write an interactive program to enter 12 rainfall values for as many sites as the user wishes, averaging the 12 for each site.
- Input: 12 rainfall amounts (`float`) for each site, ‘y’ or ‘n’ answer to prompt for continuing (`char`)

Problem Solving Case Study

- Outputs: The average (`float`) for each site, displayed to two decimal places.
- Discussion: Several loops are required: one to process the data from all sites, one to sum the 12 values at each site, two more loops for data validation
- Assumption: The user processes data for at least one site.

Problem Solving Case Study

```
//*************************************************************
// Rainfall program
// This program inputs 12 monthly rainfall amounts from a
// recording site and computes the average monthly rainfall.
// This process is repeated for as many recording sites as
// the user wishes.
//*************************************************************

#include <iostream>
#include <iomanip>    // For setprecision()

using namespace std;

void Get12Amounts( float& );
void GetOneAmount( float& );
void GetYesOrNo( char& );
```


Problem Solving Case Study

```
int main()
{
    float sum;           // Sum of 12 rainfall amounts
    char  response;      // User response ('y' or 'n')

    cout << fixed << showpoint          // Set up floating pt.
         << setprecision(2);            //   output format
    do
    {
        Get12Amounts(sum);
        cout << endl << "Average rainfall is " << sum / 12.0
             << " inches" << endl << endl;
        cout << "Do you have another recording site? (y or n) ";
        GetYesOrNo(response);
    } while (response == 'y');
    return 0;
}
```

Problem Solving Case Study

```
void Get12Amounts( /*out*/ float& sum ) // Sum of rainfall

// Inputs 12 monthly rainfall amounts, verifying that
// each is nonnegative, and returns their sum
// Postcondition:
//     12 nonnegative rainfall amounts have been read
//  && sum == sum of the 12 input values
{
    int   count;       // Loop control variable
    float amount;      // Rainfall amount for one month

    sum = 0;
    for (count = 1; count <= 12; count++)
    {
        cout << "Enter rainfall amount " << count << ": ";
        GetOneAmount(amount);
        sum = sum + amount;
    }
}
```


Problem Solving Case Study

```
void GetYesOrNo( /*out*/ char& response )  // User response char

// Inputs a character from the user and, if necessary,
// repeatedly prints an error message and inputs another
// character if the character isn't 'y' or 'n'
// Postcondition:
//     response has been input (repeatedly, if necessary, along
//     with output of an error message)
//  && response == 'y' or 'n'
{
    do
    {
        cin >> response;
        if (response != 'y' && response != 'n')
            cout << "Please type y or n: ";
    } while (response != 'y' && response != 'n');
}
```


Problem Solving Case Study

```
void GetOneAmount( /*out*/ float& amount )   // Rainfall amount
                                             // for one month
// Inputs one month's rainfall amount and, if necessary,
// repeatedly prints an error message and inputs another
// value if the value is negative
// Postcondition:
//     amount has been input (repeatedly, if necessary, along
//     with output of an error message)
//  && amount >= 0.0
{
    do
    {
        cin >> amount;
        if (amount < 0.0)
            cout << "Amount cannot be negative. Enter again: ";
    } while (amount < 0.0);
}
```