IIT Bombay

Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Iterative Programs: Putting It All Together

Quick Recap of Relevant Topics

- Iteration idioms in programming
- “while …”, “do … while …” and “for …” loops in C++
- Use of assignment expressions and their variants in loops
- Use of “break” statements in loops

Overview of This Lecture

IIT Bombay

- Putting things together
  - Writing programs with iteration using what we have learnt
- Use of “break” and “continue” statements

Putting It All Together: An Example

- Given positive integer inputs “m” and “n”, calculate $3^{min(m, n)}$ and $2^{max(m,n)}$, and print their values

C++ Program for Min/Max Example

```
int main() {
  int minMN = 0, maxMN = 0, twoRaisedMax = 1, threeRaisedMin = 1;
  int m, n, i, j;  // Inputs and running counters
  cout << “Give m and n: “;    cin >> m >> n;
  for (i = m, j = n; ((i >= 1) || (j >= 1));  i--, j--)   // Iterate max(m, n) times
    {  if ((i >= 1) && (j >= 1)) { minMN++;  threeRaisedMin *= 3; }
       maxMN++;   twoRaisedMax *= 2;
    }
    cout << “2^max is: “ << twoRaisedMax << “, 3^min is: “ << threeRaisedMin;
   return 0;
}
```

C++ Program for Min/Max Example

IIT Bombay

```
int main() {
  int minMN = 0, maxMN = 0, twoRaisedMax = 1, threeRaisedMin = 1;
  int m, n, i, j;  // Inputs and running counters
  cout << “Give m and n: “;    cin >> m >> n;
  for (i = m, j = n; ((i >= 1) || (j >= 1));  i--, j--)   // Iterate max(m, n) times
    {  if ((i >= 1) && (j >= 1)) { minMN++;  threeRaisedMin *= 3; }
       maxMN++;   twoRaisedMax *= 2;
    }
    cout << “2^max is: “ << twoRaisedMax << “, 3^min is: “ << threeRaisedMin;
   return 0;
}
```

C++ Program for Min/Max Example

IIT Bombay

```
int main() {
  int minMN = 0, maxMN = 0, twoRaisedMax = 1, threeRaisedMin = 1;
  int m, n, i, j;  // Inputs and running counters
  cout << “Give m and n: “;    cin >> m >> n;
  for (i = m, j = n; ((i >= 1) || (j >= 1));  i--, j--)   // Iterate max(m, n) times
    {  if ((i >= 1) && (j >= 1)) { minMN++;  threeRaisedMin *= 3; }
       maxMN++;   twoRaisedMax *= 2;
    }
    cout << “2^max is: “ << twoRaisedMax << “, 3^min is: “ << threeRaisedMin;
   return 0;
}
```

C++ Program for Min/Max Example

```
int main() {
  int minMN = 0, maxMN = 0, twoRaisedMax = 1, threeRaisedMin = 1;
  int m, n, i, j;  // Inputs and running counters
  cout << “Give m and n: “;    cin >> m >> n;
  for (i = m, j = n; ((i >= 1) || (j >= 1));  i--, j--)   // Iterate max(m, n) times
   {  if ((i >= 1) && (j >= 1)) { minMN++;  threeRaisedMin *= 3; }
      maxMN++;   twoRaisedMax *= 2;
   }
   cout << “2^max is: “ << twoRaisedMax << “, 3^min is: “ << threeRaisedMin;
   return 0;
}
```

C++ Program for Min/Max Example

IIT Bombay

```
int main() {
  int minMN = 0, maxMN = 0, twoRaisedMax = 1, threeRaisedMin = 1;
  int m, n, i, j;  // Inputs and running counters
  cout << “Give m and n: “;    cin >> m >> n;
  for (i = m, j = n; ((i >= 1) || (j >= 1));  i--, j--)   // Iterate max(m, n) times
   {  if ((i >= 1) && (j >= 1)) { minMN++;  threeRaisedMin *= 3; }
      maxMN++;   twoRaisedMax *= 2;
   }
   cout << “2^max is: “ << twoRaisedMax << “, 3^min is: “ << threeRaisedMin;
  return 0;
}
```

Conditionally iterate min(m,n) times

C++ Program for Min/Max Example

IIT Bombay

```
int main() {
  int minMN = 0, maxMN = 0, twoRaisedMax = 1, threeRaisedMin = 1;
  int m, n, i, j;  // Inputs and running counters
  cout << “Give m and n: “;    cin >> m >> n;
  for (i = m, j = n; ((i >= 1) || (j >= 1));  i--, j--)   // Iterate max(m, n) times
    {  if ((i >= 1) && (j >= 1)) { minMN++;  threeRaisedMin *= 3; }
       maxMN++;   twoRaisedMax *= 2;
    }
    cout << “2^max is: “ << twoRaisedMax << “, 3^min is: “ << threeRaisedMin;
   return 0;
}
```

C++ Program for Min/Max Example

```
int main() {
  int minMN = 0, maxMN = 0, twoRaisedMax = 1, threeRaisedMin = 1;
  int m, n, i, j;  // Inputs and running counters
  cout << “Give m and n: “;    cin >> m >> n;
  for (i = m, j = n; ((i >= 1) || (j >= 1));  i--, j--)   // Iterate max(m, n) times
    {  if ((i >= 1) && (j >= 1)) {
          minMN++;  threeRaisedMin *= 3; // Conditionally iterate min(m,n) times
        }
        maxMN++;   twoRaisedMax *= 2;   // Executed max(m, n) times
    }
    cout << “2^max is: “ << twoRaisedMax << “, 3^min is: “ << threeRaisedMin;
   return 0;
}
```

Putting It All Together: Quiz Marks Problem

Read quiz 1 marks of CS101 students one at a time

Any negative marks other than -1000 is invalid and must be ignored

Marks of -1000 denotes end of input

Compute number of valid marks, sum, average, minimum and maximum

C++ Program for Quiz Marks Problem

```
int main () {
  int marks, sum, count, min, max;
  float average;
  for (sum = 0, count = 1;     ; count++)  {
     cout << “Give marks of student “ << count << “: “;  cin >> marks;
     if (marks < 0) { if (marks != -1000) { cout << “Invalid marks!!!” << endl; … Repeat loop …}
                           else { cout << “All marks read!!!” << endl; break; }
     }
     // Update sum, min, max
  }
     average = sum/(count + 0.0);
     cout << “Count of valid marks: “ << count  << “Sum: “ << sum;
     cout << “Average: “ << average << “Min: “ << min << “Max: “ << max << endl;
     return 0;
}
```

C++ Program for Quiz Marks Problem

IIT Bombay

```
int main () {
 int marks, sum, count, min, max;
 float average;
 for (sum = 0, count = 1;    ; count++)  {
    cout << “Give marks of student “ << count << “: “;  cin >> marks;
    if (marks < 0) { if (marks != -1000) { cout << “Invalid marks!!!” << endl; … Repeat loop …}
                          else { cout << “All marks read!!!” << endl; break; }
    }
    // Update sum, min, max
 }
    average = sum/(count + 0.0);
    cout << “Count of valid marks: “ << count  << “Sum: “ << sum;
    cout << “Average: “ << average << “Min: “ << min << “Max: “ << max << endl;
    return 0;
}
```

Skip all instructions in body of loop in this iteration and **exit loop**

“continue” statement in C++ Program

IIT Bombay

Skip all instructions in body of loop in this iteration and **start next iteration of loop**

```
int main () {
  int marks, sum, count, min, max;
  float average;
  for (sum = 0, count = 1;     ; count++)  {
     cout << “Give marks of student “ << count << “: “;  cin >> marks;
     if (marks < 0) { if (marks != -1000) { cout << “Invalid marks!!!” << endl; continue; }
                              else { cout << “All marks read!!!” << endl; break; }
     }
     // Update sum, min, max
 }
     average = sum/(count + 0.0);
     cout << “Count of valid marks: “ << count  << “Sum: “ << sum;
     cout << “Average: “ << average << “Min: “ << min << “Max: “ << max << endl;
     return 0;
}
```

C++ Program for Quiz Marks Problem

```
int main () {
 int marks, sum, count, min, max;
 float average;
 for (sum = 0, count = 1;     ; count++)  {
   cout << “Give marks of student “ << cou
   if (marks < 0) { if (marks != -1000) { c
                  else { cout << “All    ks re
   }
   // Update sum, min, max
 }
   average = sum/(count + 0.0);
   cout << “Count of valid marks: “ << count  << “Sum: “ << sum;
   cout << “Average: “ << average << “Min: “ << min << “Max: “ << max << endl;
   return 0;
}
```

```
sum = sum + marks;
if (count == 1) { min = marks; max = marks; }
else {
  min = (min > marks) ? marks: min;
  max = (max < marks) ? marks: max;
}
```

C++ Program for Quiz Marks Problem

```
int main () {
  int marks, sum, count, min, max;
  float average;
  for (sum = 0, count = 1;     ; count++)  {
     cout << “Give marks of student “ << count << “: “;  cin >> marks;
     if (marks < 0) { if (marks != -1000) { cout << “Invalid marks!!!” << endl; continue; }
                               else { cout << “All marks read!!!” << endl; break; }
     }
     // Update sum, min, max
 }
     average = sum/(count + 0.0);
     cout << “Count of valid marks: “ << count  << “Sum: “ << sum;
     cout << “Average: “ << average << “Min: “ << min << “Max: “ << max << endl;
     return 0;
}
```

Summary

- Examples of two C++ programming problems
- Iterative constructs with assignment expressions