Lab 1 Solutions

6.096 Staff

1 "Hello, World" (10 points)

1.1 Hello World I (1 point)

```cpp
#include <iostream>

using namespace std;

int main()
{
    const char* str = "Hello, World!";
    cout << str << "\n";
    return 0;
}
```

1.2 Hello World II (9 points)

1. `for` loop: (3 points)

```cpp
#include <iostream>

using namespace std;

int main()
{
    int N;
    cin >> N;
    for(; N-- > 0;)
    {
        cout << "Hello, World!\n";
    }
    return 0;
}
```

2. `while` loop: (3 points)

```
#include <iostream>

using namespace std;

int main()
{
    int N;
    cin >> N;
    while(N-- > 0)
    {
        cout << "Hello, World!\n"
    }
    return 0;
}
```

3. `do...while` loop: (3 points)

```
#include <iostream>

using namespace std;

int main()
{
    int N;
    cin >> N;
    do
    {
        cout << "Hello, World!\n";
    }
    while(--N > 0);
    return 0;
}
```

2 More Programs (50 points)

2.1 Scope (10 points; 2 points each)

1. We cannot declare the same name within a block because it will generate a compiler error.
2. The program compiles.
3. The declaration in the inner block is used.
4. The declaration in the outer block is used.
5. The code will not compile because the function `cout` has not yet been defined. If we move `#include <iostream>` to the top, then the code will compile.

2.2 Basic Statistics (10 points)

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int N;
8      cout << "Enter N: ";
9      cin >> N;
10     int acc = 0;
11
12     // handle the first number separately
13     cin >> acc;
14     int minVal = acc;
15     int maxVal = acc;
16
17     // then process the rest of the input
18     for(int i = 1; i < N; ++i)
19     {
20         int a;
21         cin >> a;
22         acc += a;
23         if(a < minVal)
24         {
25             minVal = a;
26         }
27         if(a > maxVal)
28         {
```

```
29              maxVal = a;
30          }
31      }
32
33      cout << "Mean: " << (double)acc/N << "\n";
34      cout << "Max: " << maxVal << "\n";
35      cout << "Min: " << minVal << "\n";
36      cout << "Range: " << (maxVal - minVal) << "\n";
37
38      return 0;
39  }
```

2.3 Prime Numbers (10 points)

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int N;
8      cin >> N;
9      for(int i = 2; N > 0; ++i)
10     {
11         bool isPrime = true;
12         for(int j = 2; j < i; ++j)
13         {
14             if(i % j == 0)
15             {
16                 isPrime = false;
17                 break;
18             }
19         }
20         if(isPrime)
21         {
22             --N;
23             cout << i << "\n";
24         }
25     }
26     return 0;
27 }
```

2.4 Multiples of numbers (10 points)

2.4.1 Ternary operator (3 points)

```cpp
#include <iostream>

using namespace std;

int main()
{
    while(1)
    {
        int N;
        cin >> N;
        cout << ((N % 5 == 0 && N >= 0) ? N/5 : -1) << "\n";
    }
    return 0;
}
```

2.4.2 continue (3 points)

```cpp
#include <iostream>

using namespace std;

int main()
{
    while(1)
    {
        int N;
        cin >> N;
        if(N % 5 > 0)
        {
            cout << "-1\n";
            continue;
        }
        cout << N/5 << "\n";
    }
    return 0;
}
```

2.4.3 break (3 points)

```cpp
#include <iostream>

using namespace std;

int main()
{
    while(1)
    {
        int N;
        cin >> N;
        if(N % 5 > 0)
        {
            cout << "-1\n";
            continue;
        }
        if(N == -1)
        {
            break;
        }
        cout << N/5 << "\n";
    }
    cout << "Goodbye!\n";
    return 0;
}
```

1 extra point if all three parts are correct.

2.5 What does this program do? (10 points)

1. Russian peasant multiplication of `bob` and `dole`. (5 points)
2. It returns 1 and exits. The operating system would assume that something went wrong. (2 points)
3. Evaluates the series

   Case $N \equiv 0, 1 \mod 4$:

   $$\frac{1}{1^2} + \frac{1}{2^2} - \frac{1}{3^2} - \frac{1}{4^2} + \frac{1}{5^2} + \cdots \frac{1}{N^2}$$

   Case $N \equiv 2, 3 \mod 4$:

   $$-\frac{1}{1^2} - \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} - \frac{1}{5^2} - \cdots \frac{1}{N^2}$$

   The sign of the $\frac{1}{N^2}$ term is positive if $N \equiv 0, 1 \mod 4$ and negative if $N \equiv 2, 3 \mod 4$. (3 points)

3 Factorials Gone Wrong (40 points)

3.1 Writing the factorial program (5 points)

0: 1; 1: 1; 2: 2; 9: 362880; 10: 3628800

3.2 Breaking the program (5 points)

If -1 is entered, the program will output 1, which is incorrect because the factorial function (or Gamma function) is not defined for negative numbers.

3.3 Breaking the program II (5 points)

The number at which the program fails depends on the system architecture; 64-bit systems often fail at 17. We accepted any answer around that value.

3.4 Rewriting Factorial (10 points)

```
1  #include <iostream>
2
3  using namespace std;
4
5  long long accumulator = 1;
6
7  int main()
8  {
9      int number;
10     cout << "Enter a number: ";
11     cin >> number;
12     if(number < 0)
13     {
14         cout << "No negative numbers allowed!\n";
15         return 1;
16     }
17     if(number > 20)
18     {
19         cout << "Program will not produce correct result!\n";
20     }
21     for(; number > 0; accumulator *= number--);
22     cout << "The factorial of " << number << " is " << accumulator
           << ".\n";
23     return 0;
24 }
```

3.5 Rewriting Factorial II (10 points)

```
#include <iostream>

using namespace std;

int main()
{
    int number;
    cout << "Enter a number: ";
    cin >> number;
    switch(number)
    {
        case 0:
        case 1:
            cout << "1\n";
            break;
        case 2:
            cout << "2\n";
            break;
        case 3:
            cout << "6\n";
            break;
        case 4:
            cout << "24\n";
            break;
        case 5:
            cout << "120\n";
            break;
        case 6:
            cout << "720\n";
            break;
        case 7:
            cout << "5040\n";
            break;
        case 8:
            cout << "40320\n";
            break;
        case 9:
            cout << "362880\n";
            break;
        case 10:
            cout << "3628800\n";
            break;
        default:
```

```
44              cout << "Input not supported!\n";
45              break;
46      }
47      return 0;
48 }
```

3.6 Further Testing (5 points)

There are no other inputs we have to consider. Since we are dealing with the short data type for storing our input number, we can separate our analysis into two cases: negative integers and nonnegative integers. We have modified our program to deal with negative numbers in 4.2 and have modified our program to deal with large positive numbers in 4.4. Therefore we do not have to consider any other input cases.

6.096 Introduction to C++

January (IAP) 2011