MA400 Computer Session Worksheet

Mark Baltovic

September 18, 2009

In this worksheet you will begin writing, compiling and executing your own programs. The idea here is not for you to finish everything in this sheet, but to gain some practical familiarity with the tools and ideas in the course. Your lab session will be supervised and you should feel free to ask them any questions you might have.

You should have read the document **MA400 Computer Session - Your Programming Environment**, which walks you through setting up your programming environment. If you have not done so, please do so now.

In this document, you will see a variety of problems for you to consider. The last problem in this worksheet is simply an introduction to functions in C++, which we shall see in the Monday lecture.

1 Summing the first n integers

We've already seen a basic version of this in the lectures. The program you will see here combines user input, simple arithmetic and a iterative loop.

Start a new source file, type in the following program, compile and then run it. Your program should prompt the user for an integer, to be stored in variable `n`, and display on the screen the sum of the integers 1 through to `n`.

```
// Prints the sum of the first n integers
#include <iostream>
#include <cstdlib>       // For EXIT_SUCCESS
using namespace std;     // Make all std names global

int main()
{
  int n;
  int sum = 0;

  cout << "Enter a positive integer : " ;
  cout >> n;

  for ( int i = 1 ; i <= n ; i = i + 1 ) {
    sum += i;
  }

  cout << "The sum of integers 1 through " << n;
  cout << " is " << sum << '\n';


  system("PAUSE");
  return(EXIT_SUCCESS);
}
```

Remark Note that you can have multiple "objects" of different types printed on the screen using one `cout` instruction.

Activities

1. See how your program reacts when you enter invalid or unexpected inputs, such as a character, string or a number with a decimal point.
2. Modify the program to try to get it to print the value of `i` after it has printed the sum. That is, *outside* the `for` loop. What happens?
3. Try removing, or commenting out (using `//`), the line

   ```
   using namespace std;
   ```

   What happens when you try to compile the program?

2 Summing over a range of integers

Using the program above, write a new program that prompts the user for two integers separately, and then returns the sum of all integers between the given integers *inclusive* of the ones given.
The integers might be entered by the user in either order: first smaller, then larger *or* first larger, then smaller. Your program should handle both cases.

Hint If you are stuck with the iterative loop, take another look at the one given in the previous section. Try to work through it, either in your head or on paper, keeping track of the values of all the variables involved: `i`, `n` and `sum`.

Does your program respond appropriately when the two integers specified are equal?

3 Writing a power program

Write a program that takes in a `double` value and an integer from the user and returns the double raised to the power of the integer.

What happens if you try to submit a non-integer power to your program?

4 Using iterative loops and arithmetic operators

Write a program that takes an integer n from the user and prints on one line, all the positive divisors of n, separated by spaces.
Modify your program so that it does this for the first n integers, with each row containing the divisors of one particular integer. Thus, your output for $n = 3$ should look like:

```
1
1,2
1,3
```

Hint Given integers n and m, we say that m is a divisor of n if and only if the remainder of dividing n by m is zero.
How could we implement an iterative loop to then check for all the divisors of an integer?

5 Writing a factorial program

Write a program that will take an integer n from the user and returns its factorial (i.e. prints to the screen).

> **Hint** Consider the definition of the factorial of an integer, and the algorithm you would use to calculate it on paper.
> How could you implement this algorithm in a program?
> Can you see how an iterative loop might help?

Modify your program so that it then takes another integer r from the user and returns the binomial coefficient:

$$\binom{n}{r}$$

What data type will you use for n and r?
Try to get a rough idea of the range of values of n that your program returns correct values.

6 The Fibonacci sequence

The Fibonacci sequence is a sequence of **integers**, defined recursively by:

$$u_1 = 1, \qquad u_2 = 1, \qquad u_n = u_{n-1} + u_{n-2} \quad n \geq 3.$$

Write a program that will take a positive integer n from the user and returns the first n members of the sequence.

Verify your results by modifying your program to check that:

a) $u_n^2 - u_{n-1}u_{n+1} = (-1)^{n-1}$.

b) $u_1 + u_2 + \ldots + u_n = u_{n+2} - 1$,

> **Remark** Notice how u_n increases very rapidly with `n` and soon exceeds the largest integer that can be represented as a fundamental type on your computer.

7 Iteration Statements: Additional exercises

Which of the iterations in the following code segments many never terminate?

a)
```
int sum = 1;
for ( unsigned i = 10 ; i >= 0 ; --i )
  sum *= 2 * i + 1;
```

b)
```
double i = 10, sum = 1;
while ( i != 0 )
  sum *= 2 * i-- + 1;
```

c)
```
int i = 0;
double sum = 1.0;
while ( 1 ) {
  sum *= 2 * i++ + 1;
  if ( i = 10 )
    break;
}
```

You should check your answers by completing these fragments as fully formed programs. (If you are having problems doing this, a template program is provided at the end of this sheet.)

8 Writing a factorial function

We have yet to see a C++ function in the lectures, but for those of you who have worked this far, and are keen to see a little more, here is a preview
Look through the following program and try to see what it is doing (the comments should help somewhat).
The program is a full listing, so it should compile and run cleanly on your computer.

```
#include <iostream>
#include <cstdlib>       // For EXIT_SUCCESS
using namespace std;     // Make all std names global

// DECLARE ancillary functions before they are used in main()
int factorial(int n);


int main()
{
  long int i;
  cout << "Enter an integer: ";
  cin >> i;

  cout << "(" << i << ")! = " << factorial(i) << '\n';
  return(EXIT_SUCCESS);
}


// DEFINE ancilliary functions
int factorial(int n)
{
  int result;
  if (n == 1 || n == 0) {
    result = 1;
  }
  else if (n < 0) {
    cout << "Error in factorial function: argument = " << n << '\n';
  }
  else {
    result = n * factorial(n -1);
  }
  return(result);
}
```

Activities

1. Two functions have been defined above:
   – what are their names?
   – what inputs, if any, do they take?
   – what outputs do they return?