

Due Thursday, 14 Oct 1999, at the beginning of class.

1 Reading

Read Sections 3.1–3.5 and 7.3 of the text. Begin reading Section 10.1.

2 Compilation Tip

Besides producing error messages, a compiler can also produce warnings about risky statements that may be erroneous. To have these warnings printed, use a command line like

To have the compiler *attempt* to check whether array accesses are out of bounds, try using

Of course, one can simultaneously use both options.

```
g++ -Wall -pedantic hello.cc
```

g++ -W

3 Problems

When submitting a program, please submit a printed copy of the program's code prepended with comments briefly describing its input and output. Please indent your code to make it readable. See Section 2.5 of the textbook for style hints. If you desire, you can also submit examples demonstrating your program's correctness.

1. (This problem is based on programming project 5 in Chapter 3 of the text.)

Write a function to compute interest on a credit-card account balance. The function takes arguments for the initial balance (a `double`), the monthly interest rate (a `double`), and the number of months for which interest is to be paid (an `int`). The value returned (a `double`) is the interest due. Interest is to be compounded monthly; that is, the first month interest is charged only on the initial balance, but in subsequent months interest is charged on the sum of the initial balance and the interest due in previous months. See Display 2.14 in the text for an example of such a calculation. (*Hint*: In your program you may want to use a `for` loop instead of a `while` loop.)

Also write a program to test your function. This program should read an initial balance, a monthly interest rate, and a number of months, and then use the function to compute the interest due and print the result with 2 decimal places. (See page 53 of the text for instructions on how to accomplish this.) It should repeat this (accepting input, and then computing and printing a result) as often as the user wishes. Choose an appropriate method for indicating end of input, and document your choice in the comments at the top of the program.

The following table shows what the function should return for some example sets of input.

| Initial balance | Monthly interest | Months | Interest due |
|-----------------|------------------|--------|--------------|
| 100 | 0.01 | 1 | 1.00 |
| 100 | 0.02 | 1 | 2.00 |
| 100 | 0.01 | 2 | 2.01 |
| 100 | 0.01 | 12 | 12.68 |

2. (a) In C++, most characters fall into one of five different classes: letters, decimal digits, whitespace characters, punctuation characters, or control characters. Write a C++ program that, given an input, prints the number of characters in each class.

Hints: See the character functions in Appendix 4. The `>>` operator ignores all whitespace characters so use the `cin.get` function. For example, if `c` is a character variable, the function call `cin.get(c)` stores the next character in `c`. Here is part of a program that reads its input, one character at a time.

```
char c;

while (cin.get(c))
{
    // Do something with the character in c.
}
```

What is the result of running the program on your program's C++ source code?

- (b) Write a program to count the frequency of all *alphabetic* characters in a file, producing a printout listing the number of a's, b's, ..., z's. The following functions may be useful. The first one converts a lowercase alphabetic character to an integer. The second function reverses the conversion.

```
// input <- lowercase alphabetic character
// output-> for an input of 'a', returns 0
//          for an input of 'b', returns 1
//          ...
//          for an input of 'z', returns 25
//          for any other input, returns an unspecified value
```

```
int charToIndex(char c)
{
    return static_cast<int>(c) - static_cast<int>('a');
}
```

```
// input <- index in the range 0, 1, ..., 25
// output-> for an input of 0, returns 'a'
//          for an input of 1, returns 'b'
//          ...
//          for an input of 25, returns 'z'
//          for any other input, returns an unspecified value
```

```
char indexToChar(int i)
{
    return static_cast<char>(i + static_cast<int>('a'));
}
```

Count both upper and lower case characters. For example, the count of a's should include both a's and A's. (Hint: Consider modifying the first function.)

Using a WWW browser, find an interesting short WWW page or file, save it to your local directory, and run your program on it, submitting the page or file and the associated counts.

4 Extra Credit

No extra credit will be given if the preceding homework problems are not solved.

1. (2 points) Rewrite the die rolling program using functions where appropriate. There is at least one place where a function would be appropriate.
2. (4 points) Modify the program to determine the average number of rolls to see all sides of an n -sided die, where n is a positive integer. For small values of n , e.g., 1 up to 20, plot the average number of rolls. What function does this look like? For example, does it look like $\log n$, n^2 , e^n ?