

Module 09: Output Formatting

*Intro to Computer Science 1 - C++
Professor Scott Frees*

Textbook

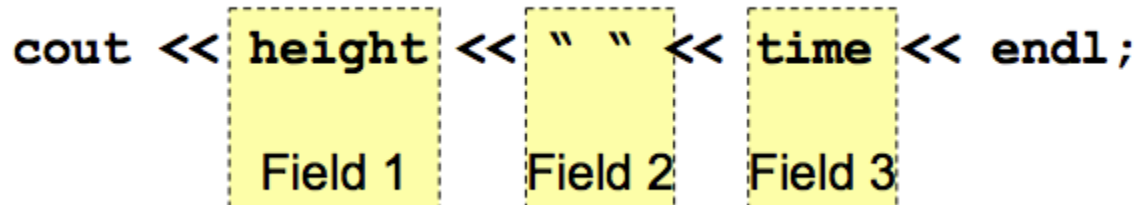
The following is covered in section 4.10 in the text (pages 140-144)

Output Formatting

- Another library: `<iomanip>`
 - Contains many “stream modifiers”
- Output Problems:
 - Set total length of “field”
 - Set the precision of decimal numbers
 - Justify field text (left or right)

Width of “fields”

```
cout << height << " " << time << endl;
```



Field 1 Field 2 Field 3

- Each field is formatted according to some default set of rules
 - ex. decimal values have up to 6 trailing digits

Width of fields

```
cout << height << " " << time << endl;
```

5		1	.	0	1	0	1	5
---	--	---	---	---	---	---	---	---

```
cout << setw(4) << height << " "  
     << setw(5) << time ;
```

			5		1	.	0	1	0
--	--	--	---	--	---	---	---	---	---

Precision

Two stream manipulators should be used to set the “significant digits”

```
cout << setprecision(1) << fixed;  
cout << setw(4) << height << " ";  
cout << setprecision(2) << fixed;  
cout << setw(5) << timeToFall;
```

	5	.	0			1	.	0	1
--	---	---	---	--	--	---	---	---	---

Justification

By default, if a field does not fill up the space set by **setw**, numbers are *right justified*.

```
cout << setw(3) << 1 << endl;  
cout << setw(3) << 10 << endl;  
cout << setw(3) << 100 << endl;
```

		1
	1	0
1	0	0

```
cout << left ;  
cout << setw(3) << 1 << endl;  
cout << setw(3) << 10 << endl;  
cout << setw(3) << 100 << endl;
```

1		
1	0	
1	0	0

Programming Example 09

Amount of money in a savings account after one year can be calculated as:

$$A = P * (1 + R/T)^T$$

A = Total Amount after one year

P = Principal (initial balance)

R = Interest Rate (Annual)

T = Times compounded

Lets print out a nicely formatted summary, including proper alignment and significant digits for dollars and cents

Lab 04

The monthly payment on a loan is calculated using the following formula:

$$\text{Monthly Payment} = \frac{\text{Rate} \times (1 + \text{Rate})^N \times L}{((1 + \text{Rate})^N - 1)}$$

Rate is the *monthly* interest rate, which is the annual interest rate divided 12. You will be asking the user to enter an *annual interest* rate (since that is how these things are usually advertised), but you need to make sure you apply this division by 12 before using that number in the equation!

N is the number of payments (or months) (i.e. if the loan term is 1 year, $N = 12$). The user will enter N directly, for example they will enter 24 if they are working with a 2 year loan.

L is the original loan amount, in dollars.

Lab 04 - cont'd

Write a program that asks the user for the **annual interest rate**, **loan amount**, and **number of payments** to be made.

Your program should compute their **monthly payment**, the **total amount** they will need to pay (monthly payment multiplied by N), and the **total interest** paid (total amount paid minus the original loan amount).

The program should echo the input out to the user and print out the results, as shown below.

Lab 04 - cont'd

The program should echo the input out to the user and print out the results, as shown below.

A sample output might be (user input in red):

```
Enter the amount of your loan:           $2000
Enter the annual interest rate (APR). (Ex. 0.09): 0.15
Enter the number of monthly payments you will make: 24
```

```
-----
Original Loan Amount:      $2000.00
Annual Interest Rate:      15.00%
Number of payments:        24
```

```
-----
Required Monthly Payment:  $96.97
Total to be paid:           $2327.36
Total interest paid:        $327.36
```