

Objectives:

- Understand memory architecture.
- Translate between binary and decimal number formats.
- Discuss the differences between, and some issues with, integer and floating-point formats.
- Create a Flow chart and use it to implement a C++ program.
- Begin exploration of basic input and output.

Binary Numbers and Memory Architecture

memory is structured in binary measurements depending on how many numbers/characters are in the amount of memory. for example is only 4bytes are used (8bits = 1 byte) you would use a long instead of a double becuase a double is used primarily for 8 bytes.

a) 11	1011	11/2 = 5.5 = 1	27/2=13.5=1	56/2=28=0	38/2=19=0
		5/2=2.5= 1	13/2=6.5=1	28/2=14=0	19/2=9.5=1
			6/2=3=0	14/2=7=0	9/2=4.5=1
b) 27	11011	2/2=1=0	3/2=1.5=1	7/2=3.5=1	4/2=2=0
		1/2=.5=1	1.5/2=.75=1	3/2=1.5=1	2/2=1=0
				1.5/2=.75=1	1/2=.5=1
c) 56	111000				

d) 38 100110 CHECK: $2^6 2^5 2^4 2^3 2^2 2^1 2^0$

a) 11011001
217

b) 00110101

53

$2^8 \ 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$
128 64 32 16 8 4 2 1

c) 10101010
170

d) 11111111

Numeric Data Types – Integers and Floating Point

4. What types of data can be represented in C++ programs?

Integer (int)
 characters (char)
 boolean (bool)
 floating points (float)
 double (double)
 function (func)
 class (class)

5. Explain how integers are represented in C++.

Short
 long
 signed
 unsigned

6. Visit the following site and read its contents:

http://www.cprogramming.com/tutorial/floating_point/understanding_floating_point.html

Explain what the author means by *Accuracy* and *Precision*. How do integers and floating-point numbers differ concerning *Accuracy* and *Precision*?

Integers and floating points are completely opposite in respects to Accuracy and Precision, integers cannot understand fractions.

7. Visit the following site and read its contents:

http://www.cprogramming.com/tutorial/floating_point/understanding_floating_point_representation.html

Based on your reading there and in the textbook, along with classroom lectures, explain how floating-point numbers are represented in C++.

floating point numbers are represented differently in every machine but is most commonly associated with IEEE-754 standard.

A sign bit = shows if number is negative or positive

e = exponent

m= mantissa (value of number is mantissa * 2^x with x being the exponent)

8. Examine the program to the right:
Look up the "isnan" definition for c++.

a) What is the function of isnan?

function of "isnan" is to return
whether or not x is a Not-a-Number
(NaN) value

NaN is used to identify undefined
or non-representable values

b) What do think that the output
of each
of the cout statements in the
program
will be?

Line 12:

x > x -> 0

Line 13:

Y>X ->1

Line 14:

Z>X ->0

Line 16:

11.1/0.0 -> inf

Line 17:

0.0/0.0 -> nan

Line 18:

log(0) -> -inf

Line 19:

sqrt(-1) -> -nan

Line 22:

a==a ->1

Line 23:

1->0.2 2.0/10.0 ->0.2

Line 24:

a == 2.0/10.0 -> 0

Line 26:

ISNAN(sqrt(-1)) ->1

Line 27:

ISNAN(5) -> 0

c) Type in the program and save it as prog2a.cpp. Compile prog2a.cpp using your compiler.
What is the

command that you issued to compile the program? If you used an IDE how did you do it?

```
1 #include<iostream>
2 #include<math.h>
3 using namespace std;
4
5 int main()
6 {
7     float x = 1.0;
8     float y = 0.0;
9     float z = 0.0;
10    y = x + 0.000001;
11    z = x + 0.000000000000000001;
12    cout << "X > X  -> " << (x>x) << endl;
13    cout << "Y > X  -> " << (y>x) << endl;
14    cout << "Z > X  -> " << (z>x) << endl;
15
16    cout << "11.1/0.0 -> " << 11.1/0.0 << endl;
17    cout << "0.0/0.0 -> " << 0.0/0.0 << endl;
18    cout << "log(0)  -> " << log(0) << endl;
19    cout << "sqrt(-1) -> " << sqrt(-1) << endl;
20
21    float a = 2.0/10.0;
22    cout << "a == a  -> " << (a==a) << endl;
23    cout << "a -> " << a << "    2.0/10.0 -> " << 2.0/10.0 << endl;
24    cout << "a == 2.0/10.0 -> " << (a == 2.0/10.0) << endl;
25
26    cout << "ISNAN(sqrt(-1)) -> " << isnan(sqrt(-1)) << endl;
27    cout << "ISNAN(5) -> " << isnan(5) << endl;
28
29    return 0;
30 }
```

I named the file main.cpp and i ran it in repl.it.com

d) Run the program you compiled above. Examine its output and compare it to the estimations you made in section (b) of this question (above). Explain any discrepancies between your estimation and the actual results. Also, explain the output for Line 14.

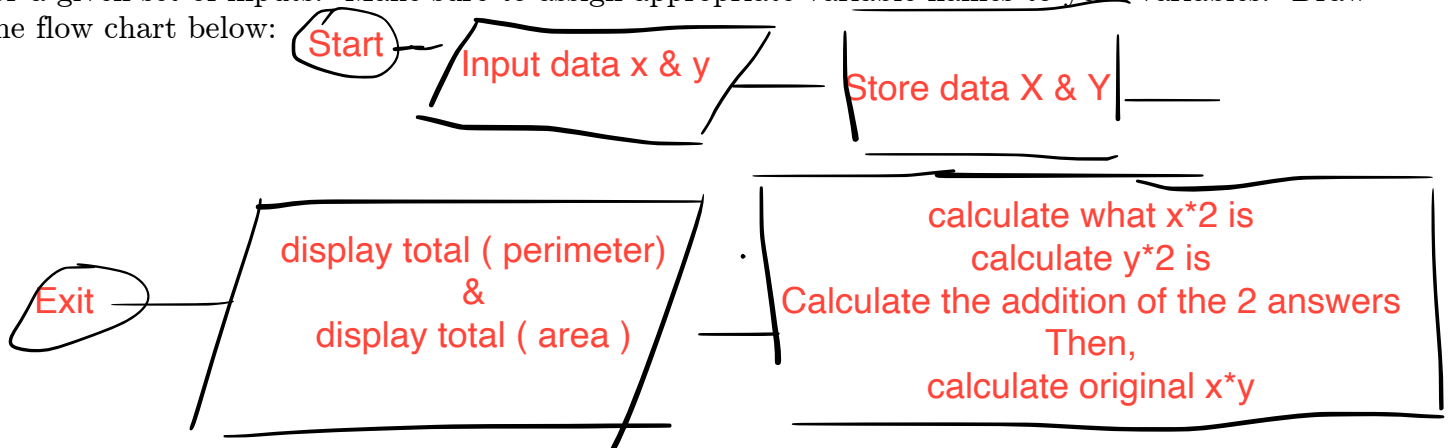
I originally wrote my assumptions but assumed it would be wrong so i ran it and filled in exactly what is was but it was pretty similar to what i originally had down except the last 2

Z is so closely related to x is rounded and gave us the answer of 0

Flow charts, program design, and implementation

10. Create a complete C++ program that lets a user calculate the perimeter and area of a rectangle given the length and width. Print the length and width, as well as the perimeter and area, to the screen.

a) Create a Flow chart to conceptualize the problem, listing the inputs, outputs, and processing steps that must occur in the program. Make sure to check your algorithm by hand tracing the output for a given set of inputs. Make sure to assign appropriate variable names to your variables. Draw the flow chart below:



b) Write a complete C++ program to implement the algorithm from your flow chart, declaring the variables you

identified in the previous step. Assign some reasonable values for the length and width.

Attach a copy of this

program to your lab sheet.

c) Compile your program.

```
#include <iostream>
using namespace std;
```

```
int main() {
```

```
    int x = 5;
```

```
    int y = 7;
```

```
    int a = x * 2;
```

```
    int b = y * 2;
```

```
    cout << (a + b) << "inches" << endl;
```

```
    cout << (x * y) << "in^2" << endl;
```

```
}
```

d) Run and check your program. Make sure the output is correct.

11. Examine the following program:

a) Type in the program. Compile the prog2b.cpp source file and run it.

b) Explain the operation of Line 8. What does *cin* do?

You can use Google to look up the answer if this was not covered in class yet.

The function of Cin is to store the data the user enters from the line before

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int x;
7     cout << "Type in an integer: ";
8     cin >> x;
9     cout << "You typed the number " << x << endl;
10
11     return 0;
12 }
```

c) Line 7 is often referred to as a prompt. Explain why we need a prompt for this program. (what would happen if we did not have one?)

We need a prompt to figure out the value of the integer (x)

If you didnt have a prompt the computer wouldnt have a value and wouldnt know what to display for line 9

d) Run the program again, and when prompted for an integer, enter a floating-point number. What is the output of the program? How/why does it differ from your input?

It was rounded down to 3, i put in 3.5

It was rounded down becuase it is being read as an integer not a float

e) Alter the area/perimeter program you wrote in section 10 of this lab. Use *cin* to read the length and width from the keyboard. Attach a copy of your modified program to this lab sheet.

```
#include <iostream>
using namespace std;
int main() {
    int x;
    int y;
    cout << "Type in value for x:";
    cin >> x;
    cout << "Type in value for y:";
    cin >> y;
    int a = x * 2;
    int b = y * 2;
    cout << (a + b) << "inches" << endl;
    cout << (x * y) << "in^2" << endl;
}
```