CSE 20 Intro to Computing I

Lecture 6 – Number Systems

Announcements

- Lab #7 this week
 - Due before your next lab
- Project #1
 - Due 10/27 (Friday)
- Midterm Exam
 - 10/23 during lecture
 - Lectures 1-6 (NOT 1-5)/Lab #1-7
 - Review during labs in week of 10/16
 - Open notes
 - No electronic devices (including calculators)
- Reading assignment
 - Chapter 4.4-4.9 of textbook

Binary -> Hex

- Groups of 4 binary bits = Hex
- Convert: 1010 1110 0110 1100
- A E 6 C
- OxAE6C

https://www.khanacademy.org/math/algebra-home/alg-intro-to-algebra/algebra-alternate-number-bases/v/binary-to-hexadecimal

Hex -> Dec/Binary

- Convert 0x5B6F
- To decimal:

$$5 \times 16^3 + 11 \times 16^2 + 6 \times 16^1 + 15 \times 16^0 = 23407_{10}$$

▶ To binary:

```
 \cdot 5_{16} = 0101_2
```

$$B_{16} = 1011_2$$

$$6_{16} = 0110_2$$

$$F_{16} = 1111_2$$

0101 1011 0110 1111₂

Numbers Conversion

- What is 5029_{10} in Binary (Base 2)?
- Find the largest power of 2 such that $2^x \le 5029$
 - \circ 12th is largest (2¹² = 4096)
 - \circ 5029 4096 = 933
- Find the largest $2^x \le 933$

$$9^{th} \rightarrow 2^9 = 512 \rightarrow 933 - 512 = 421$$

Find the largest $2^x \le 421$

$$\circ$$
 $8^{th} \rightarrow 2^8 = 256 \rightarrow 421 - 256 = 165$

Find the largest $2^x \le 165$

$$0 \mid 7^{th} \mid \rightarrow 2^7 = 128 \rightarrow 165 - 128 = 37$$

Find the largest $2^x \leq 37$

$$\circ$$
 $5^{th} \rightarrow 2^5 = 32 \rightarrow 37 - 32 = 5$

Find the largest $2^x \le 5$

$$2^{n} \longrightarrow 2^2 = 4 \longrightarrow 5 - 4 = 1$$

Find the largest $2^x \leq 1$

$$0^{th} \rightarrow 2^0 = 1 \rightarrow 1 - 1 = 0$$

Put 1's in each power of 2 that we found OR 0's in others

Binary: 1 0011 1010 0101

3 A 5

Hex: 0x13A5

Check:

$$1 \times 16^3 + 3 \times 16^2 + 10 \times 16^1 + 5 \times 16^0$$

 $\rightarrow 4096 + 768 + 160 + 5 = 5029$

https://www.khanacademy.org/math/algebrahome/alg-intro-to-algebra/algebra-alternatenumber-bases/v/large-number-decimal-to-binary

Numbers: Operations

Symbol	Function		
*	Multiply		
/	Divide		
%	Remainder		
+	Add		
-	Subtract		

Numbers: Comparison

Operator	Meaning	
==	equal	
!=	not equal	
<	Less than	
>	Greater than	
<=	Less than equal	

The result is either True or False

Numbers: logical operator &&(AND)

Input 1	Input 2	Expression	Output
0	0	0 && 0	0
0	1	0 && 1	0
1	0	1 && 0	0
1	1	1 && 1	1
Input 1	Input 2	Expression	Output
FALSE	FALSE	False AND False	FALSE
FALSE	TRUE	False AND True	FALSE
TRUE	FALSE	True AND False	FALSE

Numbers: logical operator | | (OR)

Input 1	Input 2	Expression	Output
0	0	0 0	0
0	1	0 1	1
1	0	1 0	1
1	1	1 1	1
Input 1	Input 2	Expression	Output
FALSE	FALSE	False OR False	FALSE
FALSE	TRUE	False OR True	TRUE
TRUE	FALSE	True OR False	TRUE
TRUE	TRUE	True OR True	TRUE

Precedence

- 1+2+3+4
 - \circ (((1 + 2) + 3) + 4)
- 1*2+3*4
 - (1 * 2) + (3 * 4)
- 1 * 2 < 3 * 4
 - (1 * 2) < (3 * 4)</pre>
- 1 < 2 && 3 < 4</p>
 - (1 < 2) && (3 < 4)
- \rightarrow (int) 1 + 2.0
 - 1+2.0 → 3.0
- \rightarrow (int) (1 + 2.0)
 - $(int)(3.0) \rightarrow 3$

- () parentheses have highest precedence
- *, /, % are next in evaluation
- +, -
- <, <=, >, >=
- ==, !=
- &&, || are last to be evaluated
- Always evaluate left to right (default)

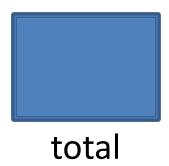
Numbers - Divide

- 7/8
 - · 0
- (float) 7 / 8
 - 0.875
- (float) (7 / 8)
 - · 0.0
- (float) 7 / 8.0
 - · 0.875
- ▶ (int) 7/8.0
 - · 0.875
- ▶ (int) (7/8.0)
 - ° 0

Numbers – Remainder (Modulus)

- 7 % 8
 - 7
- 8 % 7
 - 1
- 2 % 1
 - · 0
- 1 % 2
 - 1
- 7 % 2
 - 1
- 8 % 2
 - · 0

int total; // Declaration



▶ int total = 0; // Declaration + initialization



- int total = 0;
- ▶ total = 1;



```
int total = 0;
```

- ▶ total = 1;
- ▶ total = 5;



```
int total = 0;
```

- ▶ total = 1;
- total = 5;
- total = 100;



```
int total = 0;
```

- ▶ total = 1;
- total = 5;
- total = 100;
- total = 0;



```
    int total = 0;
    total = 1;
    total = 5;
    total = 100;
    total = 0;
    total = total + 1;
```



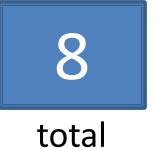
```
int total = 0;
total = 1;
total = 5;
total = 100;
total = 0;
total = total + 1;
total += 5;
```



```
int total = 0;
total = 1;
total = 5;
total = 100;
total = 0;
total = total + 1;
total += 5;
total++;
```



```
int total = 0;
total = 1;
total = 5;
total = 100;
total = 0;
total = total + 1;
total += 5;
total++;
++total;
```



```
int total = 0;
total = 1;
total = 5;
total = 100;
total = 0;
total = total + 1;
                                              total
total += 5;
total++;
++total;
System.out.println(total++== 9); false ... Why?
```

Increment after comparison

```
int total = 0;
total = 1;
total = 5;
total = 100;
▶ total = 0;
total = total + 1;
total += 5;
total++;
++total;
System.out.println(++total == 9); true
```

9

total

Project (Bobcar.java)

```
Available cars: 1 for Economy, 2 for Compact, 3 for Standard
Please choose the rental car: 1
Please enter the number of rental days: 3
Club member?: 1 for yes, 0 for no: 1
Platinum Executive Package?: 1 for yes, 0 for no: 1
                                                 $ 105
Base: 3 days for a Economy @ $35 per day:
Club Member Discount:
                                               - $ 0
                                               + $ 15.75
Platinum Executive Package:
                                                 $ 120.75
Total Estimate for Rental:
```

Approaching the Project

- Read carefully
- Step by step
 - Need 3 (sometimes 4) main inputs from user
 - Print out the results from variables before you do anything
 - Calculate a base cost for each car
 - Add in discount for Club Member
- Premium Package
 - add input only if club member
 - logic to calculate and print out
- Put this at the end of main
 - input.close(); // For Resource leak warning. input is the name of scanner