

# Discussion 8

# Project 4 Hints

- linegraph and insertionsort can have an unlimited number of user inputs

- concept of “keep getting until...” something entered
  - for the project “...until blank line entered”
  - ONE solution for this (many exist):

```
while (TRUE)
```

```
{
```

```
    string input = GetLine(); //similar to GetInteger, etc., this gets full line
```

```
    if (StringCompare(input, "") == 0) //if both arguments same, fn returns 0
```

```
        break;
```

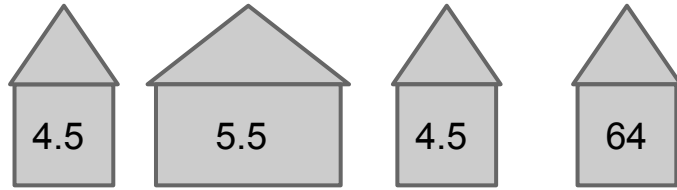
```
    //if no break occurred, do work with input
```

```
}
```

# Pointers

- House analogy:
  - very variable is a house on a road
  - contents of the house is the value of the variable
  - address of the house is the memory address of the variable

# Pointers - example 1 (pointers1.c)



	X	Y	Z	W
House Address:	64	76	80	84

& - address of  
\* - value of (value of variable pointed to)

\*\*\*\* full example is in code: pointers1.c

```
float x; //house x created
double y; //house y created
x = 4.5;
y = 5.5;
printf("%f\n",x); //prints 4.5
printf("%d\n",&x); //prints 64 (in my computer)
printf("%d\n",%y); //prints 76 (in my computer)
float z = x;
printf("%f\n",z); //prints 4.5
printf("%d\n",&z); //prints 80 (in my computer)
float * w=&x; (no * prompts compiler error)
printf("%d\n",w); //prints 64
printf("%d\n",&w); //prints 84 (in my computer)
printf("%f\n",*w); //prints 4.5
```

# Pointers - example 2 (pointers2.c)

- Pointers and arrays can be thought of interchangeably.
- Please see code and notice the print statements