Goals

Friday, January 26, 2018 9:54 AM

Expand on Pointers

- non-native Types (struct)
- the -> operator

Functions:

Pass-by-value

- reference
- pointer
- array

Dynamic Variables

Pass-by-Value Functions

Friday, January 26, 2018 9:58 AM

```
void passByVal (int a)
```

- when called, "copy" of the argument's value
- e.g.

```
int main( ) {
    int x = 3
    pBV(x)
    cout << x <<endl
}
```

```
pBV (int a)
{
    a++;
}
```

local variable

3

Pass-by-Ref

```
void myPBR( int &a)
```

- e.g.

```
int main(  ) {
    int x = 3;
    pBR(x)
}
```

```
myPBR (int &a)
{
    ++a;
}
```

4

Pass By Pointer

```
void myPBP(int *p)
```

- like w/ PBR, an address is being passed. However, have to call w/ pointer type.

-e.g. call:

```
int *myPtr;
myPBP(myPtr)
```

-e.g. call

```
int x;
myPBP(&x);
```

-e.g. ~~*x~~

```
int x;
int *xPtr;
myPBP(xPtr);
```

Pass-by-array

```
void myPBA(int arr[], int n)
```

-like with PBR and PBP, an address is being passed

- the whole array is <u>NOT</u> passed, just a pointer to the 1st element