

COMP364 Lecture 14

Messages:

- HW2 due Wednesday, late homework due Friday.

Today:

- Array index
- More example problems
- Hashes?

Array indexing:

- from last class...
arrays can be viewed as a 'list' or as 'scalar' numbers.

```
@Arr=('ted', 'ann', 'jerry', sam')
      0      1      2      3
```

```
> print $Arr [2];      (when talking about the whole array, we use the '@' sign. When referring to a
                        specific component of the array eg. [2], we use the '$' sign).

=> jerry
```

eg.

```
$Name = $Arr[1];      ($Name is 'ann').
$Arr[3] = 'Frank';     (where @Arr is ('ted', 'ann', 'jerry', 'frank'))
$Arr[6]='mary';        (@Arr is ('ted', 'ann', 'jerry', 'frank', , , , mary'))
```

example problem:

Print contents of array, one per line:

```
$count=0;
while ($count<@Arr)
{
    print $Arr[$count]\n ;
    $count++;
}
```

example problem 2:

Print only even elements of array:

```
$count=0;
while ($count<@Arr)
{
    if (($count%2)==0)
    {
        print Arr[$Count]\n ;
    }
    $count++;
}
```

... alternative answer:

```
$count=0;
while ($count<@Arr)
{
    print $Arr[$count]\n ;
    $count+=2;
}
```

'Indices' to access >1 element at a time

```
@A=(ted, jerry, Frank, 92, F, 6, tom, 10);
@A2=@A[1,3,5];    ('@' used instead of '$' as we are accessing a sub array, and assigning it to A2).
=> @A2 is (jerry, 92, 6)
```

```
@A3=@A[2..5];
=> @A3 = (Frank, 92, F, 6)
```

– **With a list of three different scalar variables:**

```
($Val1, $Val2, $Val3) = @A
=> $Val is ted
    $Val2 is jerry
    $Val3 is Frank
```

eg.
@A = (a,b,c,d,e,f)
\$Num = @A;
=> \$Num is 6, as we are assigning the list as scalar values.

```
@B=(x,y,z);
$B[1] = @A;
=>B is (x, 6, z)
```

```
@B[2] = @A;
=> B is (x,y, a)      ('@' is present, so we treat it as a list of variables, thus we take the list
                        of values and assign them one by one).
```

```
$B[10]=sam;
=> (x,y,a,b,c, , , , , , sam)
```

example problem:

Suppose you are given an array @A, scalar \$B. How many times does the \$B value appear in @A?

```
$count=0;
foreach $Val (@A)
{
    if ($Val eq $B) {      ('eq' used instead of '==' as it is a string comparison, not for numbers)
        $count++;
    }
}
```

```
print $count\n ; (see course website for full code)
```

example problem 2: with array indexing?

```
$count2=0;
$count=0;
$len = @A; (where $len is the string for the length of the array)
while ($count<$len)
{
    if ($A[$count] eq $B)
    {
        $count2++;
    }
    $count++;
}
print $count2 ; (see course website for full code)
```