Recitation 4: Pointers, Arrays, Strings

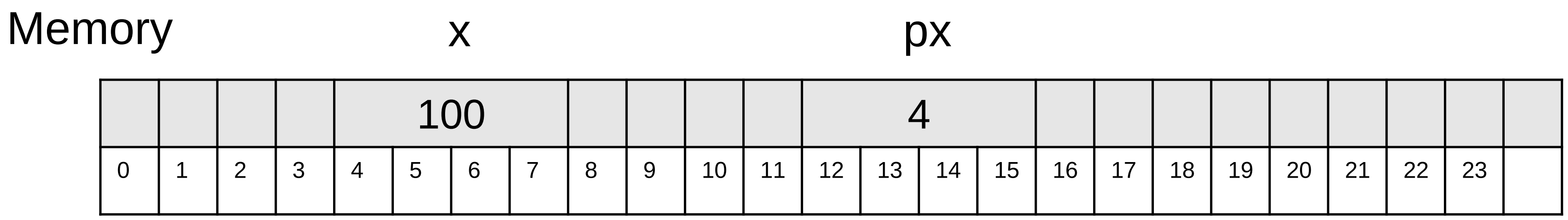
Thursday, February 12th, 2015

Pointers (1/2)

- ***Variable:*** represents a memory location that stores data (size vary according to datatype).
- ***Pointer:*** is a variable whose value is an address of a memory location (size of a pointer is always the same on a system: **4 bytes** on a 32 bit machines, and **8 bytes** on a 64 bit machine)

- ***Basics: (Don't get confused!)***
  - ***&*** Address Operator : &<variable>
  - **_*_** Dereferencing Operator : * <pointer>

Pointers (2/2)



```
Int x;
Int* px;
```

```
px = 4
  x = 100
  &x : ?
*px : ?
```

```
  *x  : ?
&px  : ?
```

code: **pointers.c**

Arrays

Declare and initialize entries immediately:

```
Int class_sections[5] = {30, 20, 40, 25, 10};
```

Declare and initialize entries later:

```
Int class_sections[5];
class_section[0] = 30; class_sections[1] = 20; …
```

Array as a pointer:

```
char my_characters[5];
*my_characters = ‘E’; *(my_characters + 1) = ‘F’; …
```

Strings

A **string** is an array of characters that is null-terminated

```
Char name[10] = {‘B’, ‘a’, ‘n’, ‘n’, ‘o’, ‘n’, ‘\0’};
```

```
Char name[10] = “Bannon”;
```

```
Char name[] = “Bannon”;
```

```
Char* name = “Bannon”;
```

What happens when the ‘\0’ char is missing?

Conclusion

- Demos
  - pointers.c
  - arrays.c (skip)
  - strings.c

Questions?