Object Referencing & Arrays

Jasleen Kaur

Fall 2011

Object-oriented (OO) languages

- Allow you to create Abstract Data Types (ADT)
  - Buzz word
    - Also referred to as “Class”
  - ADT have:
    - Name
    - Variables (hold values/info that represents the state)
    - Operations (defined as functions)
      - Often set/get values of state variables
  - Can create instances of an ADT/Class – “Object”
Object instances created with `new`

- `x = new Employee();`
- `y = new Employee();`

Each instance has its own copy of data and functions

- `x.setName("Bob");`
- `y.setName("Sue");`

Predefined class – String

An example of a pre-defined class – String:

- `var s = "abcd";`
- `var t = "123456";`

- `s, t`: objects (instantiations) of the String class

- Can invoke operations like: `s.length`, `s.substring`, ...

- `var c = s.charAt(0);`
- `var l = s.substring(1, s.length);`
- `var x = t.charAt(4);`
Let's take a slight detour

- What does this code do?

function swap(i, j)
{
    var temp = i;
    i = j;
    j = temp;
}

... 
var x = 10;
var y = 20;
swap(x, y);
alert("x is " + x + ", y is " + y);

What does this code do?

function swap(a, i, j)
{
    var temp = a[i];
    a[i] = a[j];
    a[j] = temp;
}

... 
var list = [1, 10, 5, 4, 3];
swap(list, 2, 4);
alert("elements are "+list[2] + ", " +list[4]);
alert(list.join()); // returns a string with all elements, comma delimited ...
Location and referencing

- Recall that variables and objects have:
  - Name, location, type, value

- In primitive types, value is stored in location
  - Numbers, boolean
  - e.g.,
    - var x = 10;
    - var y = 20;
    - var ans = true;

- When these are used as parameters
  - swap(x, y);
  - You get copies of value
    - Copies go away as soon as function returns

Object storing and referencing

- Objects are stored differently
  - var e1 = new Employee();

- Arrays are objects
  - var list = new Array();

- When used as parameters
  - swap(list, i, j);
  - You get copies of values
    - Function still works on same array!
  - You can not change actual parameter
    - But you can change something referenced by a parameter