Linked List (I): Preparation

CSCD 300 – Data Structures

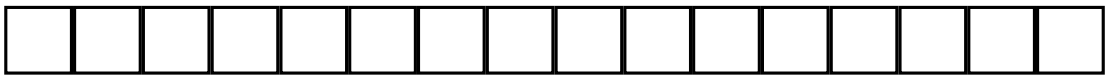
Eastern Washington University

EASTERN WASHINGTON UNIVERSITY
start something big

Outline

Recall: The characteristics of RAM

- The RAM can be viewed as a linearly collected memory storage cells.
- Every cell has a unique address, which continuously increases.
- Any unit cell can be accessed directly using contant time (a.k.a. random access) by knowing the address of that cell, no matter where the storage cell locates.



RAM

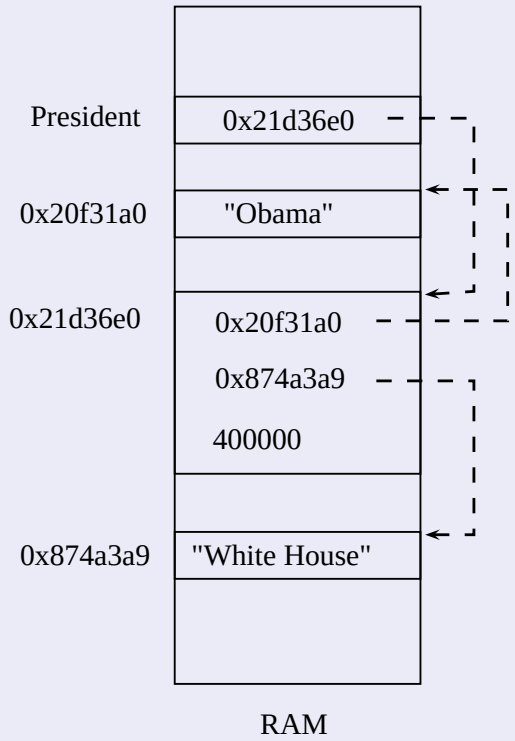
Java's reference mechanism

The Java programming language provides the programmer with a mechanism called reference to obtain the address of a RAM cell and access the data stored there [1].

An example

```
class Employee{
   String name;
   String place;
   int salary;
   /*constructor is here. */
}
Employee President = new Employee("Barack",
                                  "White House",
                                  400000);
```

By using the variable `president`, we will be able to obtain the reference that points to the RAM location that stores the actual object. Of course, inside of the object, there could be other references pointing to other RAM locations that store the object's fileds that are not of primitive type.



[1] Other programming languages such as C and C++ also provide such a mechanism under different names.

Java supports recursive class definition [2]

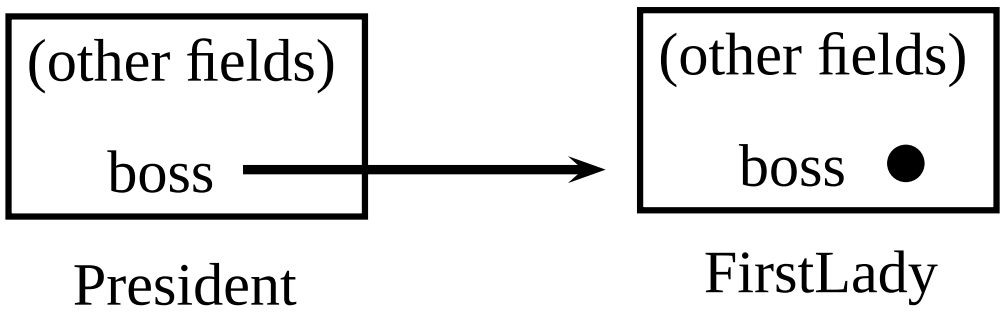
```
class Employee{
   String name;
   String place;
   int salary;

   Employee boss; /* The reference to an object
                     of the type being defined ! */

   /*constructor is here. */
}
```

[2] Other programming languages such as C and C++ also support similar mechanism.

Create two connected objects

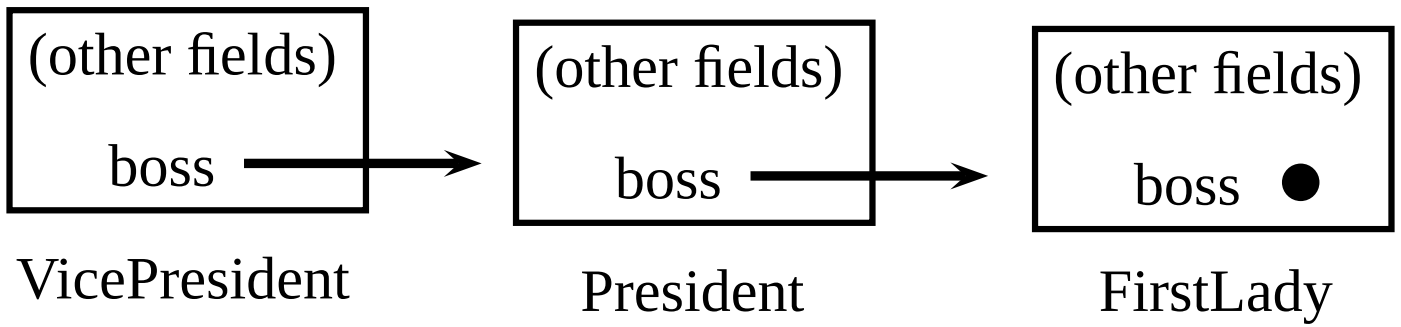
```
Employee FirstLady = new Employee("Michelle", "White House",
                                  0, null);
Employee President = new Employee("Barack", "White House",
                                  400000, FirstLady);
```

Then, given the variable `President`, we can use `President.boss` to obtain the reference pointing to the object which is being referenced by the variable `FirstLady`. Logically, we are having two linked objects of the same type:



Create three connected objects

```
Employee VicePresident = new Employee("Biden", "DC",
                                      231900, President);
```



The concept of Singly Linked List

- All the three objects connected together at the previous page is overall called a linked list of three nodes.
- Each object is represented by a node in the linked list.
- By using the reference pointing to the head node, which in the particular example in the previous page is the variable `VicePresident`, we are able to visit all the nodes in the linked list by following the `boss` link, hop by hop.
- This linked list is a singly linked list, because each node has only one "next-hop" link.

Java code example

The code shown here is only for a simple demonstration, meaning the code is not well designed from the software engineering's point of view. In the next lecture, we will demonstrate a formal Java implementation of a singly linked list by using the same idea that we have discussed.

```
public class Employee{
    public String name;
    public String place;
    public int salary;
    public Employee boss;

    public Employee(String name, String place,
                    int salary, Employee boss){
       this.name = name;
       this.place = place;
       this.salary = salary;
       this.boss = boss;
    }
}
```

```
public class test_simple_LL{
    public static void main(String[] args){
        /* Of course you can use other ways to build the linked list
           by assigning object to another object as its boss */
        Employee FirstLady = new Employee("Michelle", "White House",
                                          0, null);
        Employee President = new Employee("Barack", "White House",
                                          400000, FirstLady);
        Employee VicePresident = new Employee("Biden", "DC",
                                              231900, President);

        for(Employee who = VicePresident; who != null; who = who.boss){
            System.out.println("Name: " + who.name);
            System.out.println("Place: " + who.place);
            System.out.println("Salary: " + who.salary);
            if(who.boss != null)
                System.out.println("Boss: " + who.boss.name);
            else
                System.out.println("Boss: " + "I am the king !");
            System.out.println();
        }
    }
}
```

Output

```
Name: Biden
Place: DC
Salary: 231900
Boss: Barack

Name: Barack
Place: White House
Salary: 400000
Boss: Michelle

Name: Michelle
Place: White House
Salary: 0
Boss: I am the King !
```