COP3502 Module 06 Review

Passing Objects as arguments (continued):

When you pass a **primitive data type variable** (like an int, double, char, etc.) as an argument to a function, Behind the scenes, a **COPY of that variable is made for the use of that function**, and then when the function ends, that COPY is deleted.

```
public static void main(String[] args)
{
        int myNumber = 3;
        myFunction(myNumber);

        System.out.println(myNumber); // 3
}

public void myFunction(int num)
{
        num = num + 1;
}
```

myNumber = 3

myFunction creates a copy of 3

copy = 3

myFunction (num = 3)

// num will update (and = 4), but when the function ends num will disappear (it falls out of scope)

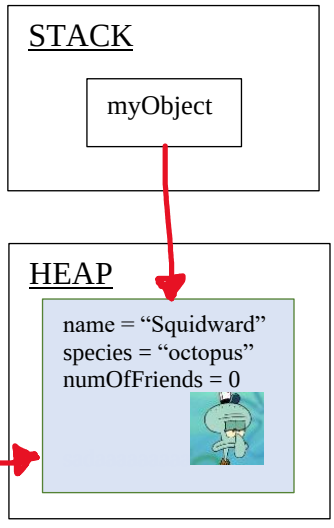
We will be left with myNumber = 3

When you pass an **OBJECT** variable as an argument to a function,
Behind the scenes you pass A **REFERENCE TO THE ACTUAL OBJECT IN MEMORY (a pointer)**
When the method receives this reference variable **IT IS POSSIBLE FOR THE METHOD TO MODIFY THE CONTENTS OF THE OBJECT** referenced by the variable.

```
public static void main(String[] args)
{
        SeaCreature myObject = new SeaCreature("Squidward" "octopus");
        myFunction(myObject);

        System.out.println(myObject.getName()); // MrKrabs
}

public void myFunction(SeaCreature obj)
{
        obj.setName("MrKrabs");
}
```



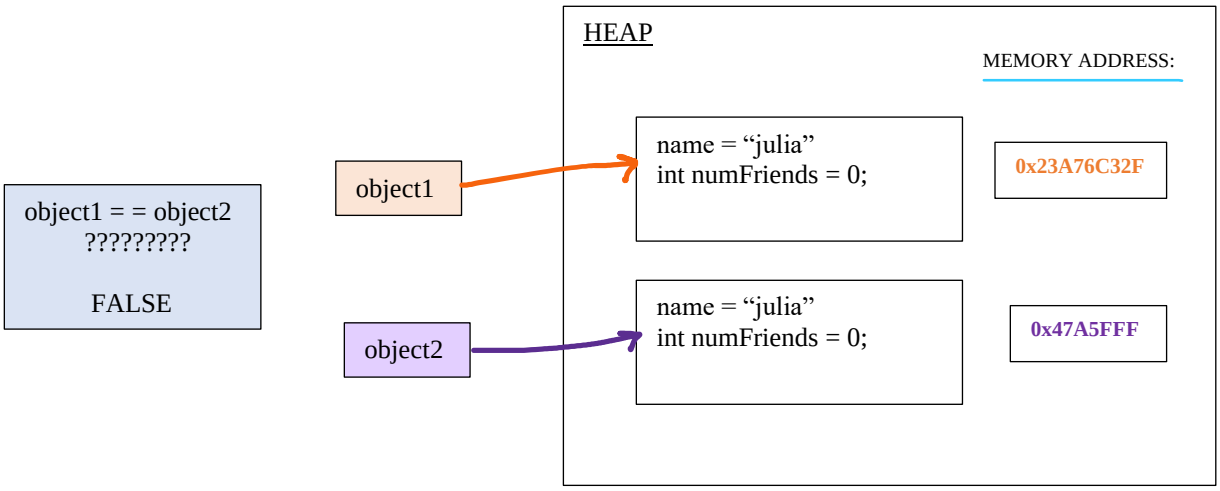
myFunction(myObject)

// the actual reference to the SeaCreature object in memory (heap) is passed to myFunction so that myFunction can make permanent changes to myObject

COP3502 Module 06 Review

Comparing objects:

When comparing objects, we cannot just use "= =". When you do something like object1 = = object2 the objects themselves will not be compared; their memory addresses will be compared.



Static variable:

A static variable (or static method) exists **in a class** and **OUTSIDE any particular object**…
A static variable of a SeaCreature class might be a count variable "numberOfSeaCreatures". This variable should not belong to any particular object (like Squidward or Spongebob) because it wouldn't make any sense. But it would make sense as a variable of the class itself (it helps us keep track of the number of SeaCreature objects we make).