Object References

- Object variables in Java are called, “object reference variables.”

- Object references are significantly different from primitive data type variables.

- To understand references, we need to go back to our diagram of memory.

- So, there are values associated with different variables.

- Let’s say we declare an object reference using the “SavingsAccount” class.

SavingsAccount acct2;

- When we call “new” and assign the result to “acct2”, what we’re going to get is a memory address for an object that is going to be created elsewhere in memory.

Diagram

Description automatically generated

- So we have this address to memory which we can think of as a reference to the actual object “acct2”.

- A reference variable's location in memory holds

A. The beginning of an object

B. A clue to another variable

C. A memory address

acct2 = 4248 (MEMORY ADDRESS)

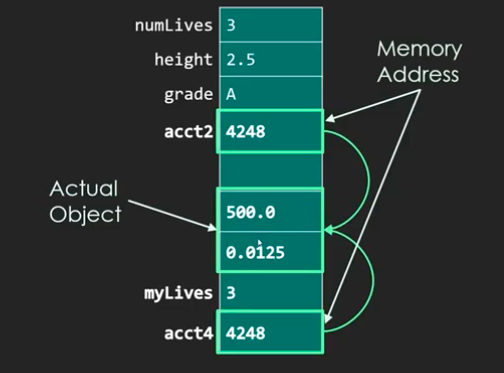
numLives = 3;

int myLives = numLives;

SavingsAccount acct4 = acct2;

- The value of “numLives” is 3 and so when I assign a new int variable “myLives” to “numLives”, the value of “myLives” is 3.

- But if I assign “acct2” to a new object called “acct4” (of the SavingsAccount class) then the value of “acct4” will be the memory address of “acct2”, which equals 4248.



- If I deposit $200 into “acct4”, I will also be depositing $200 into “acct2” because they have the same memory address.

int num1, num2

student(class) stu1, stu2

num1 = num2;

num2++;

stu1 = stu2;

stu2.setName(“John Jones”);

// setName changes the name stored in the Student object.

- We have four variables: num1 and num2 are integers, and stu1 and stu2 are of type Student (which is a class), all of which have been appropriately declared and initialized with data. Suppose our program has the following lines of code: num1 = num2; num2++; stu1 = stu2; stu2.setName("John Jones"); // setName changes the named stored in the Student object. Which of the following is guaranteed to be true after the code is executed?

1. num1 and num2 have both increased by 1, and stu1 and stu2 both have the name "John Jones"
2. Only num2 has increased by 1, while stu1 and stu2 both have the name "John Jones"
3. num1 and num2 have both increased by 1, but only stu2 has the name "John Jones"
4. Only num2 has increased by 1, and only stu2 has the name "John Jones"

- Explanation: An assignment of a primitive variable to another variable (both of a compatible data type) will not change the variable itself, it will just set the assigned variable equal to the value of the other. But for object references, an assignment of a particular object reference to another compatible object reference will set the memory address to both of the object references. “Such as acct2 and acct4.”

- ‘new’ is a Java keyword. It never needs to be defined since it’s part of the language.

Check the ObjectReferencePractice.java file for the actual code.