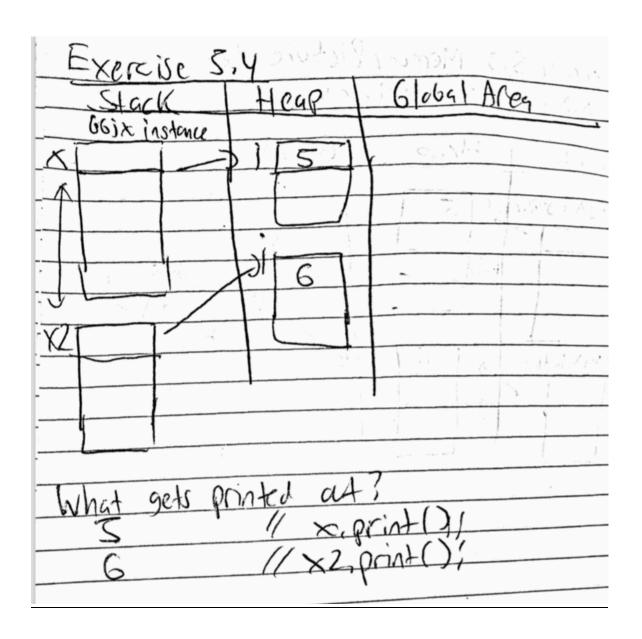
<u>Exercise 5.2</u> Go back to the first example, <u>DynamicExample.java</u>, and draw the memory pictures for this example at various points in the execution of main().

~			
	Memory (	Pictur amic <u>Example. 94</u> V   Global Arca	
Exercise	S.2 luc Dur	ianic Example 19V	91
Stack	Llaco	Clobal Arra	7
	4974	610041 15101	
Ocix instance			
1001× 110.11	3 5		
			1
X 4974 CCCO			
100	9386		+
			+-
GCO			+
	Min 6cco		
	10 600		,
	$\mathcal{U} \setminus \mathcal{V}$		
	//		
	/		
	()	· 1 · · · · · · · · · · · · · · · · · ·	
PONT	s: i=S		
F1773. 123			
<u>J= 6</u>			
	•		

Exercise 5.3: Consider the example below and draw the memory picture just after the execution of x2.i=6

Exercise 5.3 Memory Picture for Dynamic & Example 9 java		
Stack	Heap	Gleba
Obj X instance  Note that the control of the contro	lov, i	ath executive C)

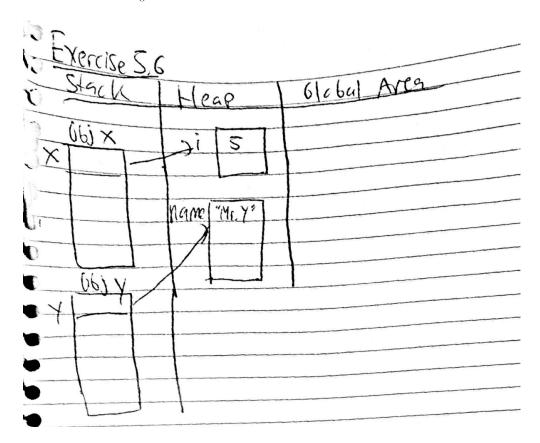
**Exercise 5.4:** Consider the example below and draw the memory picture just after the execution of x2.i=6



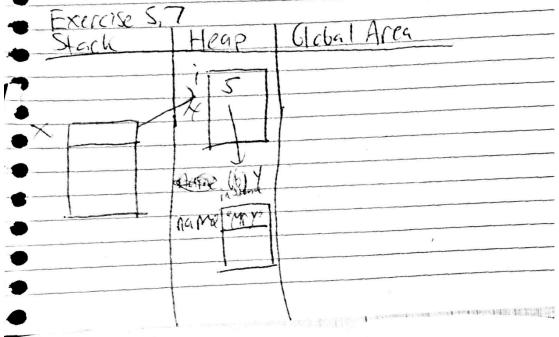
**Exercise 5.5:** What is the memory picture just after executing the only line in printTheObject(), but before the method returns?

EXERCISE SHACK  SHACK  Make Anobject  A particular of the shace  A particul	Heap Globa  (B)  (B)  (B)  (C)  (D)  (D)  (D)  (D)  (D)  (D)  (D		
1) = 06jxx= make Ancique (); (a) = 06jx 06j = new 0(jx () 1B) = 06j, T=5! return 06j! (atter sets 06jX x historice eq			
	10 06j in 9(069)		
2) Print th 2A) Obj. prin	reabject (ObjX obj)		

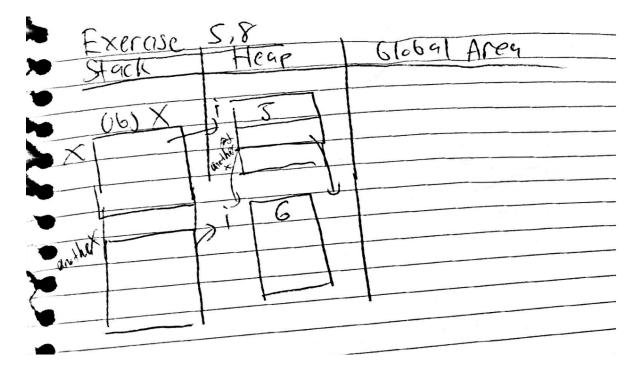
Exercise 5.6: Download <u>DynamicExample8.java</u> and draw the memory picture just after the last statement in main().



Exercise 5.7: Draw the memory picture after the last line executes in main().



Exercise 5.8: Draw the memory picture after the last line executes in main().



Exercise 5.9: Download the <u>above program</u>, compile and run. Now remove the toString() method from ObjX, compile and run. What do you see?

Before removing the toString() method from ObjX it prints:

i=5

Object x: i=5

After removing the toString() method from ObjX it prints:

ObjX@7852e922

Object x: ObjX@7852e922

It prints the memory address which is what toString() does by default instead of the value of i when you do not override the default toString() method.

## Exercise 5.10:

Declare the two variables inside your class Complex as static. Does it work? Explain.

Yes, declaring the variables for realPart and imagPart as static still works because you are creating an instances of complex in the main method, if you were not creating instances in the main method then declaring the variables would not work (they would need to be dynamic objects)

Exercise 5.11: (in folder with code)

**Exercise 5.12:** Consider these two familiar lines of Java code:

System.out.println ("Number of primes below 100: "); System.out.println (25);

• What is out? Is it a class? A method?

System is a class that is included in the java.lang package. "out" is a static member of this class, it is in particular an instance of java.io.PrintStream. It is not a class or a method but a variable that is used to to initialize the PrintStream constructor.

• Where do you find the println() method? What is its signature?

Just like "out", println() is a method found in java.io.PrintStream in the java.lang package. Its signature is println(), the default return type is void so it has no parameters. However, every time you use it, you are overloading the default method in order to print what you put in the parentheses using the

Exercise 5.13: (code in folder)