Chad Philip Johnson
CSCI20, Worthington
October 09th, 2012

Pencil & Paper 09

1. Thoroughly describe what a constructor is in Java.
Our textbook offers two definitions for a constructor: 1) "A special variety of method that is designed to initialize instance variables of an object when it is created" (226), and 2) "A variety of method that is called when an object of the class is created using *new*. Constructors are used to initialize objects. A constructor must have the same name as the class to which it belongs. Arguments for a constructor are given in parentheses after the class name" (232). Some additional properties of constructors are that they have no return type, do not include *void* in the constructor heading, and can do almost anything normal functions do but are generally used for initializing an object. Also of importance is that Java always creates a constructor with no arguments if a class is lacking a constructor definition.

2. Write the code to create a new object named horse of the class Animal.
Animal horse = new Animal("neigh", "glue");

3. What is the difference between a constructor and a set method, in terms of changing the state (for example, changing the value of an instance variable) of an object?
A constructor initializes variables for an instance of the class as it is created. The set methods of an instance of a class change the values of these variables after the object has been created.

4. What is the specific term for the constructor being used to create the object chicken below?

```
        Animal chicken = new Animal();
```

The no-argument constructor (also called a no-arg constructor) is used in the creation of the object *chicken*. Any class without a constructor has a no-arg constructor created automatically by Java. However, a class with only a single or many argument constructors do not have a no argument constructor created automatically: a new instance of a class that is not passed an appropriate number arguments will result in an error. Our textbook offers the following definition of no argument constructors: "A constructor with no parameters. If [a] class definition contains absolutely no constructor definitions, then java will automatically create a no argument constructor. If [a] class definition contains one or more constructor definitions, then java does not automatically generate any constructor; in this case, what is defined is what you get" (236).

5. Can you create an object without using a constructor, such as the following?

```
        Tractor myTractor;
```

This simply creates the variable *myTractor* which is of type *Tractor.* This variable is able to contain instances of *Tractor* and the instances of any subclasses of *Tractor*. It does not actually create an instance of the class Tractor.

6. Is the following invocation of a constructor correct (assume the code is using the object chicken that was already created above)? Explain your answer.

```
        chicken.Animal("moo", "milk");
```

No. A constructor is only invoked when the instance of a class is created. It cannot be used afterward. According to our textbook, "A constructor is called when you create a new object, such as with the operator *new*. An attempt to call a constructor in any other way… is illegal" (233).

7. Can a constructor call other methods that are defined with the same class? For example, could the default constructor for Animal below call setNoise as a part of its actions?
Yes, other methods that are part of the same class can be invoked from within a constructor. According to our textbook, "It is perfectly legal to invoke another method within the definition of a constructor…. This is legal because the first action taken by a constructor is to automatically create an object with instance variables. You do not write any code to create this object. Java creates it automatically when the constructor is invoked. Any method invocation in the body of the constructor definition has this object as its calling object" (234).

8. Does the class Animal below have a default constructor? Explain your answer.
Yes, it does have a default constructor, which appears in the given code. A "default constructor" is simply another term for a "no argument constructor." According to our textbook, "The term default constructor is misleading because… a no argument constructor is not always provided by default" (236).

9. Does the class Barn below have a default constructor? Explain your answer.
Yes, it does have a default constructor, even though one does not appear in the given code. Although a constructor has not been defined explicitly within the class, Java automatically creates a "no argument" or "default" constructor. According to our textbook, "If you define a class and include absolutely no constructors of any kind, then a no-argument constructor is automatically created. This no-argument constructor does not do much, but it does give you an object of the class type" (235).

10. Does the class Tractor below have a default constructor? Explain your answer.
No, it does not have a default constructor. Because a single argument constructor has been defined, Java does not create a "no-argument" or "default" constructor. Java only creates a default constructor when no constructors have been defined in the class.