CSSE 220 – Object-Oriented Software Development
Rose-Hulman Institute of Technology

Worksheet 04

Name (Print):_____ Section:_____

1. **Class & Object Terminology**

Term	Notes from lecture
class	
object	
instance	
field	
method	

2. **Creating an Object Instance**. Suppose you have a class `Block`. Complete the code to create and use an object.

```
Block dirt = _________________________________;   // create object
dirt.type = "Dirt";
dirt.hardness = 1;
```

3. Suppose you have an object called **stone** and the class Block defines a method called **mine()**. Now you need to call the mine method.

```
1   Block stone = new Block();
2
3   ____________________ // // call the mine() method
4
```

4. **What is a Constructor?**

A constructor is _____

5. **Constructor Rules**

- The constructor name must match the class name (T / F)
- A constructor has no return type (T / F)
- You always must write a constructor (T / F)
- A Class can have multiple constructors (T / F)

6. **The Default (No-Arg) Constructor**
   If no values are provided, fields receive Java's default values:

   - String → _____
   - int → _____
   - Object → _____

7. **Using `this` in a Constructor**. Complete the constructor:

```
1  public class Block {
2      String type;
3      int hardness;
4
5      public Block(String type, int hardness) {
6
7          ------------------------------
8
9          ------------------------------
10     }
11 }
```

8. **Why private?**. Give one reason why a class would make a field private:

   _____

9. **Accessing Private Fields**. To allow controlled access to the private field, we use (circle the correct one):

   - a **getter / setter** to read the field: _____
   - a **getter / setter** to change the field: _____

10. **True / False**
    A top-level class in Java can be declared `private` (T / F)

11. **Applying Access Modifiers**. Rewrite the class so that `type` is public and `hardness` is private.

```
class Block {
    __________ String type;
    __________ int hardness;
}
```

12. **Before You Leave**
    Write one question you still have about classes, objects, or constructors.