FPP Quiz 2/16/2021

1.
```
class MyClass {
    public static void main(String[] args) {
        myMethod();
    }

    public void myMethod() {
        System.out.println("hello");
    }
    }
```

When you compile/run this program the result is:

a. Outputs `"hello"` to the console
b. Compiler error
c. Runtime exception

2.
```
class MyClass {
    public static void main(String[] args) {
        MyClass m = new MyClass();
        m.myMethod();
    }
    private void myMethod() {
        System.out.println("hello");
    }
    }
```

When you compile/run this program the result is:

a. Outputs `"hello"` to the console
b. Compiler error
c. Runtime exception

3. Suppose `MyClass` and `AnotherClass` are in the same package.

```
class MyClass {
        public static void main(String[] args) {
            AnotherClass a = new AnotherClass(new MyClass());
            a.anotherMethod();
        }
        void myMethod() {
            System.out.println("hello");
        }
    }

    class AnotherClass {
        MyClass m;
        AnotherClass(MyClass m) {
            this.m = m;
            anotherMethod();
        }
```

```
        void anotherMethod() {
              m.myMethod();
        }

  }
```

When you compile/run this program the result is:

a. Outputs `"hello"` to the console
b. Outputs `"hello"` *twice* to the console
c. Compiler error
d. Runtime exception

4.
```
class MyClass extends MySuperClass {
  public static void main(String[] args) {
        MySuperClass cl = new MyClass();
        System.out.println(cl.getType());
  }

  public int getType() {
        return 3;
  }
  }
  class MySuperClass {
        public int getType() {
              return 2;
        }
  }
```

What happens when the program is compiled/run?

a. Compiler error
b. Runtime error
c. Outputs 2 to the console
d. Outputs 3 to the console

5.
```
class MyClass extends MySuperClass {
        public static void main(String[] args) {
              MySuperClass cl = new MySuperClass();
              System.out.println(cl.getType());
        }

        public int getType() {
              return 3;
        }
  }
  class MySuperClass {
        public int getType() {
              MyClass cl = new MyClass();
              c1.getType();
```

```
                return 2;
            }
        }
```

What happens when the program is compiled/run?

a. Compiler error
b. Runtime error
c. Outputs 2 to the console
d. Outputs 3 to the console

6.
```
class TheClass {
    TheClass(){
        TheSubclass sub = new TheSubclass();
        System.out.println("The Class constructor");
    }
    public static void main(String[] args){
                new TheSubclass();
    }
}
class TheSubclass extends TheClass {
    TheSubclass() {
        System.out.println("The Subclass constructor");
    }
}
```

What happens when the program is compiled/run?

a. Compiler error
b. Outputs "The Subclass constructor" followed by "The Class constructor".
c. Outputs "The Class constructor" followed by "The Subclass constructor".
d. An Exception occurred

7. What is the state of `e1` and `e2` after the constructor of `Test` exits?

```
public class Test {
    Employee e1 = new Employee("mike", 5000);
    Employee e2 = new Employee("joe", 8000);
    void modify(Employee x, Employee y) {
        x.setName(y.getName());
    }
    public static void main(String[] args) {
        new Test();
    }
    Test(){
        modify(e1,e2);

    }
}
```

```
public class Employee {

    private String name;
    private double salary;

    public Employee(String name, double salary){
        this.name = name;
        this.salary = salary;
    }
    //getters and setters not shown here...


}
```