FPP Quiz 1

1.
```
class MyClass{
   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) {
      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

3.
```
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

4.
```
class MyClass {
   public static void main(String[] args) {
      AnotherClass a = new AnotherClass(new MyClass());

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

class AnotherClass {
   AnotherClass(MyClass m) {
      m.myMethod();
   }
}
```
When you compile/run this program the result is:

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

5.
```
class MyClass {
   public static void main(String[] args) {
      AnotherClass a = new AnotherClass(new MyClass());
   }
   private void myMethod() {
      System.out.println("hello");
   }
}

class AnotherClass {
   AnotherClass(MyClass m) {
      myMethod();
   }
}
```
When you compile/run this program the result is:

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

6. 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

7. 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");
      a.anotherMethod();
   }
}

class AnotherClass {
   MyClass m;
   AnotherClass(MyClass m) {
      this.m = m;
   }
   void anotherMethod() {
      System.out.println("hello");
      m.myMethod();
   }
}
```

When you compile/run this program the result is:

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

8.
```
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

9.
```
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