DESIGNING A TAXONOMY

UMD CS DEPARTMENT

1. Overview

In this lab you will be designing a series of classes that use Java's inheritance mechanism to demonstrate a simple "mock-up" of various kinds of cell phones. We have provided a driver for you to test your implementation as you code in the `SimpleMain.java` file. Its output should look like:

```
Charlie is receiving a call from Fred

Fred is receiving a call from Cindy's text phone
Betsy's text phone is receiving a call from Cindy's text phone
Betsy's text phone has received TEXT from Cindy's text phone:What r u doing?

Susan's camera phone now displaying picture of FunnyPic-Susan's camera phone
has received TEXT from Pete's camera phone:LOL.
Betsy's text phone has received TEXT from Susan's camera phone:ROFL
Fred is receiving a call from Susan's camera phone
Susan's camera phone now displaying picture of Fred-Susan's camera phone is
receiving a call from Fred

Charlie is receiving a call from Fred
Charlie is receiving a call from Cindy's text phone
Charlie is receiving a call from Susan's camera phone
```

(With different line breaks, however; we've edited the output slightly to fit into this display space.) We have also provided some `PublicTests` that will be used by the Submit Server in computing your grade.

1.1. Things to be done ...

Obviously, you will need to replace the bodies of the methods that contain `RuntimeException` code with the logic specified by the comments in the method bodies. In addition, you will need to find and provide the missing constructors. As often is the case in taxonomies, subclasses depend upon the existence of constructors on their parent classes in order to initialize variables that are unavailable to the target classes because they are usually `private` on their parent classes.