

## Exception Examples

Complete this table. In the "Checked?" column, identify each exception as "checked" or "unchecked". Look at the Java API docs for help.

**Checked Exception** - an exception that you are required to handle. Java requires you to use "try - catch" around the code or write "throws xxxException" on method.

**Unchecked Exception** - an exception that you are not required to handle.

| Exception                      | Checked?  | Example   |
|--------------------------------|-----------|---|
| NullPointerException           | unchecked |   |
| ClassCastException             |           |   |
| ArrayIndexOutOfBoundsException |           |   |
|                                |           | <pre>List&lt;String&gt; list = new ArrayList&lt;&gt;( ); list.add("foo"); String foo = list.get(2);</pre>                           |
| NumberFormatException          |           |   |
|                                |           | <pre>Scanner scanner = new Scanner("four"); int n = scanner.nextInt( );</pre>   |
|                                |           | <pre>Scanner scanner = new Scanner("four"); String a = scanner.next(); String b = scanner.next( );</pre>                            |
|                                |           | <pre>FileReader fr = new FileReader(     "doesnotexist.foo");</pre>   |
|                                |           | <pre>// create String for money value double value = 10.0; String s = String.format("%d Baht", value);</pre>                        |
|                                |           | <pre>// What exception <u>should</u> be thrown? Purse purse = new Purse(-1); // or here: purse.insert( new Coin(-1,"Baht") );</pre> |

2. Find the possible exceptions in this equals method.

```
public class Person {
    private String name;
    private LocalDate birthday;
    /** initialize a new Person object. */
    public Person(String name, LocalDate bday) {
        this.name = name;
        this.birthday = bday;
    }

    public boolean equals(Object obj) {
        Person other = (Person)obj;
        return this.birthday.equals(other.birthday)
            && this.name.equals(other.name);
    }
}
```

At each line of the equals method, what exception could be thrown and what is the cause of exception?

```
Scanner console = new Scanner(System.in);
System.out.print("Do you want homework? ");
String reply = console.next();
if (reply == "yes") System.out.println("Read all of Big Java");
else if (reply == "no") System.out.println("lazy!");
else System.out.println("Invalid reply.");
```

3. When we run the program the student types "yes":

Do you want homework? **yes**

What does the program print next? Why?

4.