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	`// Done in class`

ClassCastException	unchecked	`public void do(Object task) {`
		`    Runnable r = (Runnable) task;`
		`    r.run( );`
		`}`
		`// String does not implement Runnable, so the cast in`
		`// do( ) will throw exception`
		`do( "Just do it" );`

ArrayIndexOutOfBounds Exception	unchecked	`int[] fib = { 1, 1, 2, 3, 5 };`
		`System.out.println( fib[10] );`
		`// "off by 1" bounds error:`
		`int n = fib.length;`
		`for(int k=2; k<=n; k++)`
		`     fib[k] = fib[k-1] + fib[k-2];`

IndexOutOfBoundsException	unchecked	`List<String> list = new ArrayList<>( );`
		`list.add("foo");`
		`String foo = list.get(2);`

NumberFormatException	unchecked	`Integer.parseInt("two")`

InputMismatchException	(*)	`Scanner scanner = new Scanner("four");`
		`int n = scanner.nextInt( );`

NoSuchElementException	(*)	`Scanner scanner = new Scanner("four");`
		`String a = scanner.next();`
		`String b = scanner.next() ;`

FileNotFoundException	checked	`FileReader fr = new FileReader(`
		`               "doesnotexist.foo");`

IllegalFormatConversion ("%d" is for integer values only)	unchecked	`// create String for money value`
		`double value = 10.0;`
		`String s = String.format("`**`%d`**` Baht", value);`

IllegalArgumentException - invalid value of parameter	unchecked	`// What exception should be thrown?`
		`Purse purse = new Purse(-1);`
		`// or here:`
		`purse.insert( new Coin(-1,"Baht") );`

(*) You have used Scanner many times, so the answer here is obvious. Did you have to use try - catch when calling scanner.next() ?

2. Find all 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;  //1
        return this.birthday.equals( //2
                other.birthday)      //3
            && this.name.equals(     //4
                other.name);
    }
```

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

Name the exceptions yourself. Here is a description of what would cause the exception:

1: obj might not be a Person reference. Illegal cast.
2: this Person's birthday might be null. (This constructor is lazy; didn't validate parameters.)
3: other might be null (this is not caught by statement //1). You can cast **`null`** to anything.
4: this Person's name might be null

Note that `other.name = null` does <u>not</u> cause an exception. If this Person's name is not null, then `this.name.equals( null )` just returns false.

```
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?

It prints "Invalid reply." The code uses == to compare String values, but the string `reply` is read from the input (hence it is a new String, not a value from the String pool), hence <u>both</u> `(reply=="yes")` <u>and</u> `(reply=="no")` are false.

(*) You have used Scanner many times, so the answer here is obvious. Did you have to use try - catch when calling scanner.next() ?