

*Read Chapter 9 from Scott
*Assignment #5 is posted.

17 Exceptions & Exception Handling

Monday, November 16, 2020 9:07 AM

What is an Exception?

```

1 public class Crash1 {
2     static int[] a = { 10, 20, 30, 40, 50 };
3     public static void main(String[] args) {
4         for (int i = 0; i < a.length; i++) {
5             System.out.println(a[i]);
6         }
7         System.out.println("Done printing the array!");
8     } // main()
9 } // Crash1

```

```

10
20
30
40
50
Done printing the array!

```

Change condition `i <= a.length`

```

10
20
30
40
50
Exception in thread "main"
java.lang.ArrayIndexOutOfBoundsException: 5
    at Crash1.main(Crash1.java:8)

```

Java Exception Handlers

```

try {
    ...some code during whose execution errors may occur...
} catch (<some exception> <variable>) {
    ...some corrective action you could perform...
}

```

Exception Handler

Example:

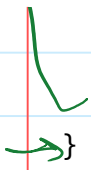
```

while (true) {
    try {
        String dataStr = JOptionPane.showInputDialog("Enter a number.");
        if (dataStr != null) {
            double data = Double.parseDouble(dataStr);
            JOptionPane.showMessageDialog(null,
                "Square root of " + data + " is " + Math.sqrt(data));
        }
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(null,
            "Bad Numeric String - try again.",
            "Input Error",




```

"Deepak"

Enter a number 32 [OK]



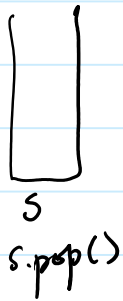
```
JOptionPane.showMessageDialog(null,  
    "Bad Numeric String - try again.",  
    "Input Error",  
    JOptionPane.ERROR_MESSAGE);
```



17 Exceptions - Terminology & Design Space

Wednesday, November 18, 2020 9:24 AM

"Cognitive overconfidence bias"



pre-defined?
user-defined?

StackError

Implicit (raised by system)
Explicit (raised by a user program)

raise/
throw

fatal?
recoverable?
clean up?

Hardware?

interrupts
software?
array index?
file not found?
divide by zero?

Exceptions

- raised/throw
- handlers

matching?
propagating?

checked?
unchecked?

Exception handlers

- ① recover + continue
- ② cannot recover, but require clean up,
- ③ nothing can be done, exit gracefully by printing message

Questions

- How + where are exceptions handlers specified? [scope]
- Can information about the exception be passed to handler?
- Does the PL allow a continuation or a termination model?

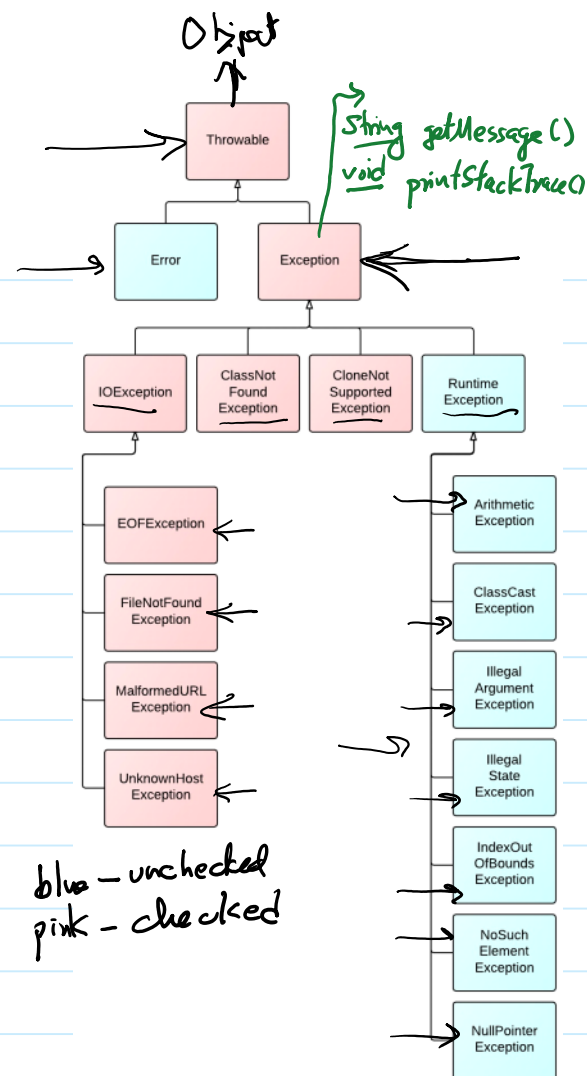
17 Exceptions in Java

Wednesday, November 18, 2020 9:24 AM

Handlers

```
try {
    // ...
} catch (<exception type> e) {
    // ...
} catch (<exception type> e2) {
    // ...
} finally {
    // always executed, if present
}
```

throw;



User-defined Exceptions

```
public class StackError extends Exception {
    public StackError (String message) {
        super (message);
    }
}
```

```
try {
    // ...
    pop() ...
} catch (StackError e) {
    // ...
}
```

```
public E pop() throws StackError {
    if (empty())
        throw new StackError ("stack Underflow.");
    // pop()
}
```

17 Exceptions in Python

Wednesday, November 18, 2020 9:24 AM

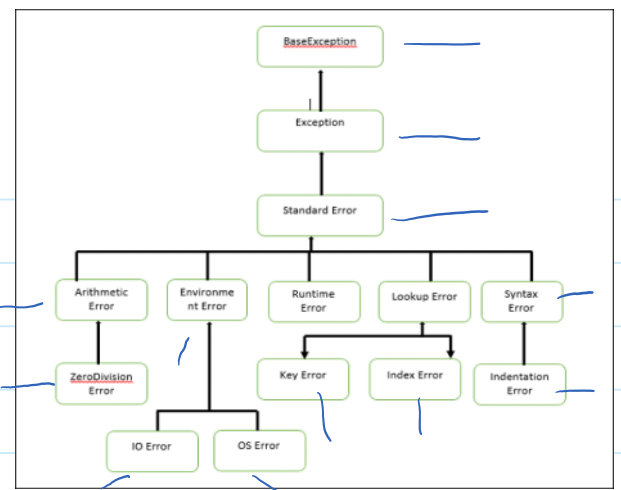
```
try:
```

```
    ...  
except e1:
```

```
    ...  
except e2:
```

```
    ...  
finally:
```

```
    ...  
else:
```



```
class StackError(Exception):
```

```
    def __init__(self, msg = "Stack Error"):
```

```
        self.message = msg
```

```
        super().__init__(self.message)
```

```
def pop(self):
```

```
    if self.isEmpty():
```

```
        raise StackError(" ")
```

```
    ...
```

```
try:
```

```
    x = s.pop()
```

```
except StackError, e
```

```
    ...
```

17 Exceptions in Go?

Wednesday, November 18, 2020 9:24 AM

error handling
panic
defer