Programming Using Java

Session 6: Exception Handling

```
public class ArrayOperators {
  public static int getRangeAverage(int[] arr, int b, int e) {
    int accum = 0;
    for (int i = b; i <= e; i++)
      accum += arr[i];
    return accum / (e-b+1);
  }
}
```

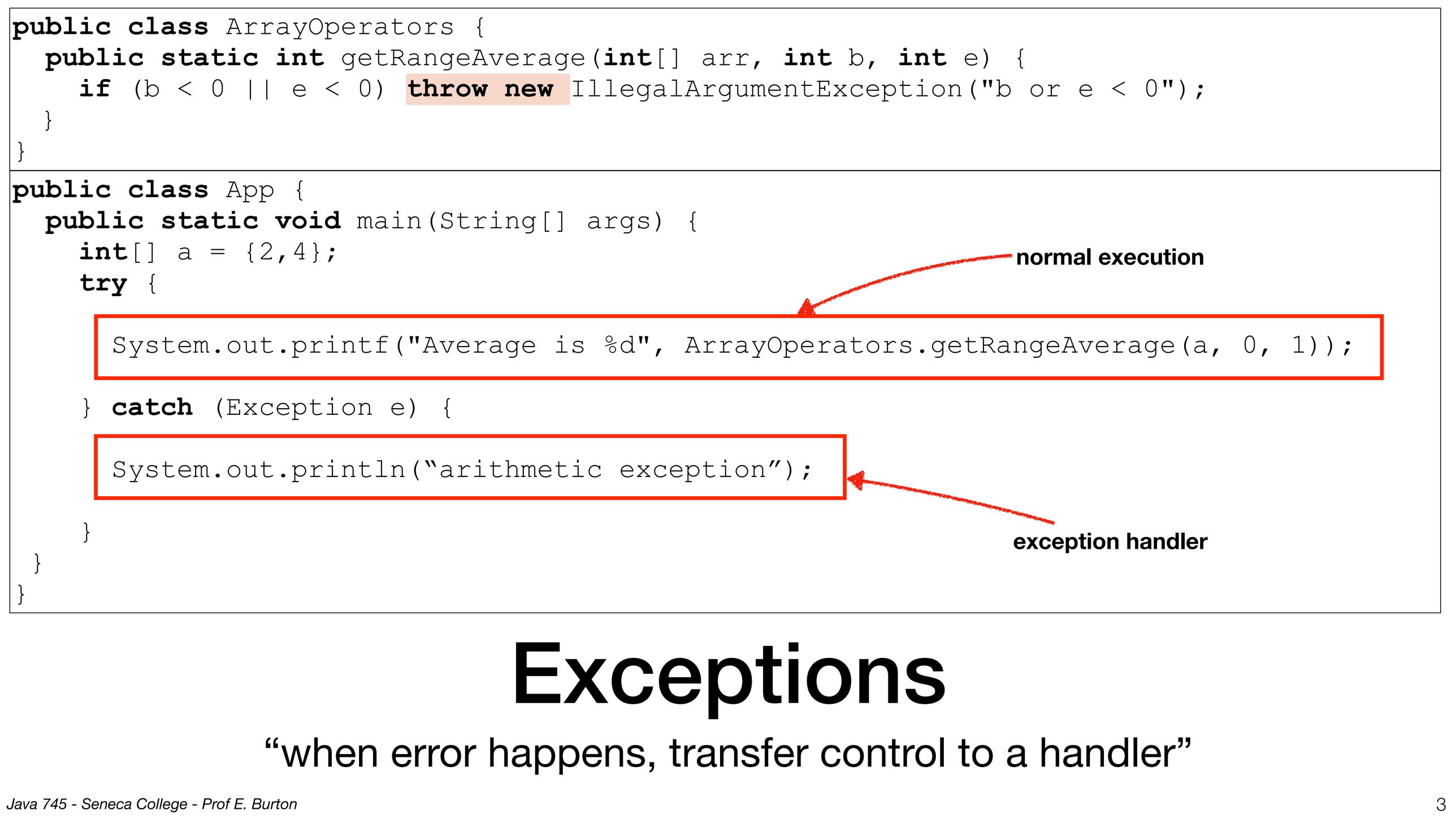
```
public class App {
  public static void main(String[] args) {
    int[] a = {2,4};
    System.out.printf("Average is %d", ArrayOperators.getRangeAverage(a, 1, 0));
 }
}
```

Error Handling

"what can go wrong?…how do we deal with it?"

```java
public class ArrayOperators {
  public static int getRangeAverage(int[] arr, int b, int e) {
    if (b < 0 || e < 0) throw new IllegalArgumentException("b or e < 0");
  }
}
```

```java
public class App {
  public static void main(String[] args) {
    int[] a = {2,4};
    try {
      System.out.printf("Average is %d", ArrayOperators.getRangeAverage(a, 0, 1));
    } catch (Exception e) {
      System.out.println(“arithmetic exception”);
    }
  }
}
```



Exceptions

“when error happens, transfer control to a handler”

```java
public class ArrayOperators {
  public static int getRangeAverage(int[] arr, int b, int e) {
    if (b < 0 || e < 0) throw new IllegalArgumentException("b or e < 0"); …
  }
}
```

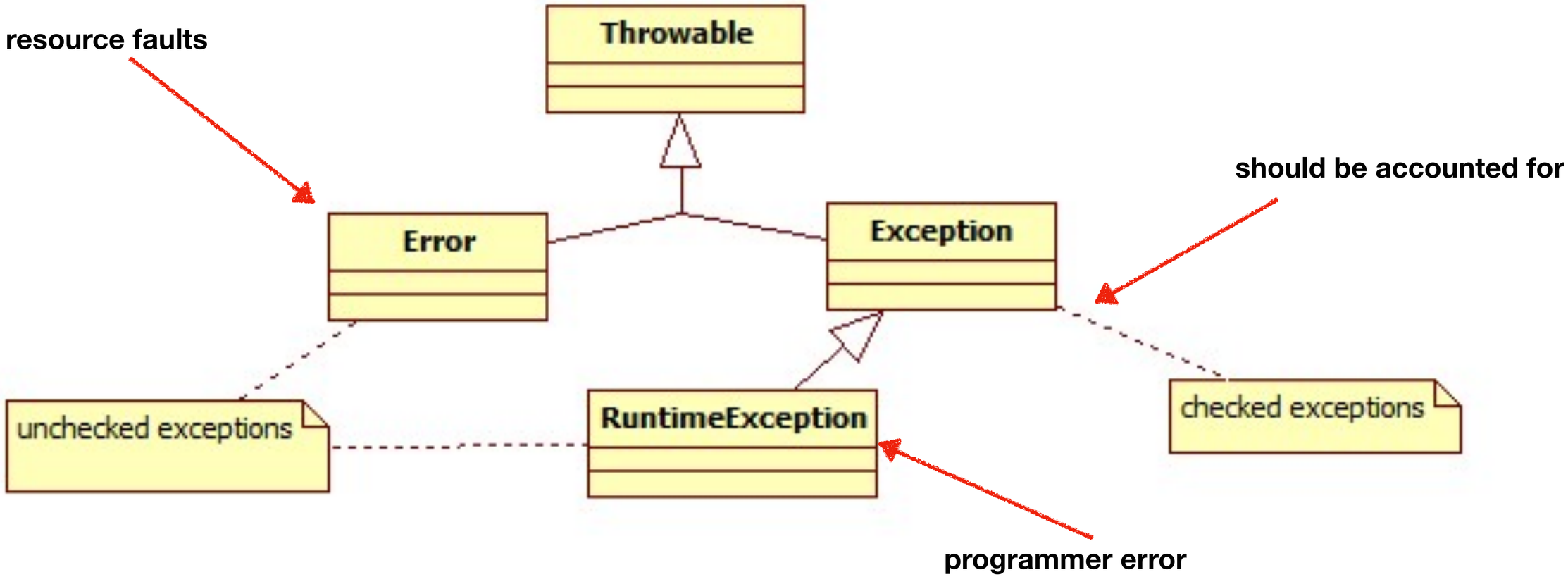
```java
public class App {
  public static void main(String[] args) {
    int[] a = {2,4};
    try {
      System.out.printf("Average is %d", ArrayOperators.getRangeAverage(a, 0, 1));
    }

    catch (IllegalArgumentException e) {System.out.println("arithmetic exception");}

    catch (Exception e) {System.out.println("arithmetic exception");}

  }
}
```

more specific handler

general handler

Exceptions

"first matching handler in executed "



Exception Class Hierarchy

"…exceptions are objects…"

let the caller handle the problem

```
/**
* @param filename
* @throws IOException
*/
public static void openFile(String filename) throws IOException {
   Formatter f = new Formatter("output.txt"); ...
}
```

handle the problem, continue execution

```
/**
* @param filename
*/
public static void openFile(String filename){
   try {
      Formatter f = new Formatter(“output.txt”);

   } catch ( IOException i) { … }
}
```

Catch or Declare

“…checked exceptions must be handled to avoid compiler errors…”

User Defined Exceptions

- helps makes errors relevant to domain
- derive from appropriate standard one
- supply overloaded constructor

```
public class FileFormatException
    extends IOException {

  public FileFormatException() {
     super();
  }
  public FileFormatException(String msg) {
     super(msg);
  }

}
```