Lec 5

Exceptions and Big-O
Exceptions

• 2 types
  – Checked exceptions
  – Runtime (unchecked) exceptions

• Errors
try
{
    // code
}
catch(ExceptionType e) {
    // code
}
public void writeArray() {
    FileOutputStream fs = null;
    PrintWriter pw = null;
    int[] a = {0,1,2};
    try {
        fs = new FileOutputStream("out.txt");
        pw = new PrintWriter(fs, true);
        for(int i = 0; i <= a.length; i++) {
            pw.println(a[i]);
        }
    } catch (Exception e) {
        System.out.println("problem with file");
    }
}

What happens if:
- We cannot open “out.txt” to write to due to permission issues
  - Program with file gets printed
  - Out.txt gets expected output
- We can open “out.txt” to write to
  - Program exits with error
  - Program exits with error
A) - Problem with file gets printed
    - Out.txt gets expected output
B) - Program exits with error
    - Program exits with error
C) - Problem with file gets printed
    - Problem with file gets printed
D) - Compiler error
E) - None of the above
public void writeArray() {
    FileOutputStream fs = null;
    PrintWriter pw = null;
    int[] a = {0,1,2};
    try {
        fs = new FileOutputStream("out.txt");
        pw = new PrintWriter(fs, true);
        for(int i = 0; i <= a.length; i++) {
            pw.println(a[i]);
        }
    } catch (IOException e) {
        System.out.println("problem with file");
    }
}
public void writeArray() {
    FileOutputStream fs = null;
    PrintWriter pw = null;
    int[] a = {0,1,2};
    try {
        fs = new FileOutputStream("out.txt");
        pw = new PrintWriter(fs, true);
        for(int i = 0; i <= a.length; i++) {
            pw.println(a[i]);
        }
    } catch (IOException e) {
        System.out.println("problem with file");
    } catch (ArrayIndexOutOfBoundsException e){
        System.out.println("array error");
    }
}
Throwing Exceptions
Methods throwing exceptions

- Methods can cause an exception, and instead of fixing it, just report to whoever called it

```java
public void writeArray() throws IOException, ArrayOutOfBoundsException {
    FileOutputStream fs = null;
    PrintWriter pw = null;
    int[] a = {0, 1, 2};
    fs = new FileOutputStream("out.txt");
    pw = new PrintWriter(fs, true);
    for (int i = 0; i <= a.length; i++) {
        pw.println(a[i]);
    }
}
```
public class Test {
    public static void main(String[] args) {
        writeArray();
    }
}

public static void writeArray() throws IOException {
    FileOutputStream fs = null;
    PrintWriter pw = null;
    int[] a = {0, 1, 2};
    fs = new FileOutputStream("out.txt");
    pw = new PrintWriter(fs, true);
    for(int i = 0; i < a.length; i++) {
        pw.println(a[i]);
    }
    pw.close();
    fs.close();
}

What happens if:
- We cannot open “out.txt” to write to due to permission issues
- We can open “out.txt” to write to

A) - exit gracefully, nothing printed
    - out.txt gets expected output
B) - program exits with run time error
    - program exits with run time error
C) - exit gracefully, nothing printed
    - exit gracefully, nothing printed
D) - Compiler error
E) - None of the above
public class Test {
    public static void main(String[] args) {
        try {
            writeArray();
        } catch (IOException e) {
            System.out.println("problem");
        }
    }

    public static void writeArray() throws IOException {
        FileOutputStream fs = null;
        PrintWriter pw = null;
        int[] a = {0, 1, 2};
        fs = new FileOutputStream("out.txt");
        pw = new PrintWriter(fs, true);
        for(int i = 0; i < a.length; i++) {
            pw.println(a[i]);
        }
    }
}

What happens if:
- We cannot open “out.txt” to write to due to permission issues
- We can open “out.txt” to write to
A (bad) solution

```java
public class Test {
    public static void main(String[] args)
        throws IOException {
        writeArray();
    }

    public static void writeArray() throws IOException {
        FileOutputStream fs = null;
        PrintWriter pw = null;
        int[] a = {0,1,2};
        fs = new FileOutputStream("out.txt");
        pw = new PrintWriter(fs, true);
        for(int i = 0; i < a.length; i++) {
            pw.println(a[i]);
        }
    }
}
```

What happens if:
- We cannot open “out.txt” to write to due to permission issues
- We can open “out.txt” to write to
终于

try {
    statements;
}

catch (TheException e) {
    handle exception;
}

finally {
    finalStatements;
}
Which statements execute?

```java
try {
    statement1;
    statement2;   //causes Exception1 to occur
    statement3;
}
catch (Exception1 e) {
    statement4;
}
finally {
    statement5;
}

statement6;
```

A) 1,2,3,4,5,6  
B) 1,2,4,5,6  
C) 1,2,5,6  
D) 1,2,4,6  
E) 1,2,6
Algorithms
What is the order of the for loop?

for(int i = 0; i<n; i++) {
    k = k+5;
}

A) O(k)  
B) O(i)  
C) O(n)  
D) O(n*constant)  
E) O(k*n)
What is the order of the for loop?

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        k = k + i + j;
    }
}

A) O(n)
B) O(n^2)
C) O(constant * n^2)
D) O(i * j)
E) None of the above
What is the order of the for loop?

```java
for(int i = 0; i<10; i++) {
    k = k + 4;
}

for(int i = 0; i < n; i++)
    for(int j = 0; j<20; j++) {
        k = k + i + j;
    }
```

A) O(n)
B) O(10 + n)
C) O(10 + 20n)
D) O(20n)
E) None of the above