CS 302 Week 15

Jim Williams, PhD
This Week

Course Evaluations
● (currently ~40%, survey closes May 5th)

Lab: Comparing Playing Cards
● using Java interfaces

Lecture: Review
● initially exceptions and "everything else".

Final Exam: Wednesday, May 10th
● location
Exception Handling

3 categories of Throwables

creating exception (checked or unchecked)

2 ways of handling exceptions

__________ checks that a checked exception is handled.
public class Test {
    public static void methodE(double value) throws MyException {
        if (value < 5.0)
            throw new MyException("value < 5.0");
    }
    public static void main(String[] args) throws MyException {
        try {
            methodE(1.5);
            System.out.println("after");
        } catch (MyException e) {
            e.printStackTrace();
            System.out.println(e.getMessage());
            //throw e;
        } finally {
            System.out.println("always done");
        }
        System.out.println("if not caught or is rethrown, not done.");
    }
}

class MyException extends Exception {
    MyException(String msg) {
        super(msg);
    }
}
If this compiles, what is true?

static void methodB(int i) {
    if (i < 3)
        throw new ExceptionQ();
}

public static void main(String[] args) {
    try {
        methodB(3);
        System.out.print("B");
    } catch (ExceptionQ e) {
        System.out.print("C");
    } finally {
        System.out.print("D");
    }
    System.out.print("E");
}
If methodA throws MyException2 then output is?

```java
public static void main(String[] args) throws MyException {
    try {
        methodA(3);
        System.out.print("B");
    } catch (MyException2 e) {
        System.out.print("C");
    } catch (MyException e) {
        throw e;
        System.out.print("D");
    }
    System.out.print("E");
}
```
Course Review

Basic fluency in Java
P1 - sequence, conditions, loops, console I/O
P2 - static methods, parameter passing, arrays
P3 - classes, instances, instance methods, exceptions, file IO

Incremental Development and Testing
Tracing code
Basic debugging - print statements, breakpoints
Eclipse IDE (Integrated Development Environment)

- Editor
- Compiler
- Translator
- Interpreter (Virtual Machine)
- Computer
- Files
- Hello.java
- Hello.class
- Users

Programmer
Programming Errors

Syntax/Compile time

runtime & Logic

Files

Compiler

Hello.java

Hello.class

(Virtual) Machine

Computer

Programmer
Memory

int k;
Integer m;

//where in memory, what values will they hold and when?
   //if local variables
   //if parameters
   //if class variables
   //if instance variables
Primitives and Wrapper classes

Draw a picture of memory and describe each.

```java
public static void main( String []args) {
    int k;
    Integer m;
    k = 2; //example of ?
    m = 3; //example of ?
    k = m; //example of ?
}
```
Strings and memory

class S {

    public static void main( String [] args) {

        String name;  // Where and what is the value?

        new String("dog");  // What and where is the value?

        name = new String("happy");  // What and where is the value?

    }

}
http://kerncountylibrary.org/event/may-the-4th-be-with-you-celebration/
Office Hours Next Week:

Tuesday, 9am to 11am
Course Evaluations

Log into www.aefis.wisc.edu using your netid
(or click on the course link from the email you were sent)

1. Go to the Notification Center and Dashboard, find course and click “Take Survey”
2. Answer questions
3. Once complete, choose “Finish and Submit”
What do these do?

```java
for ( char ch = 'A'; ch < 'Z'; ch += 3) {
    System.out.print( ch);
}
```

```java
ArrayList<Bug> bugs = new ArrayList<>();
bugs.add( new Bug());
for ( int i = 0; i < bugs.size(); i++) {
    bugs.add( new Bug());
}
```
What is print out?

```java
static void myPrint( int out) {
    System.out.println( "i: " + out);
}
static void myPrint( String s) {
    System.out.println( "s: " + s);
}
.... in main....
int i = 5;
myPrint( i + "3" + (3 + 4 * 6) + 4);
```
class Dag {
    static int smop;
    Hup h;
    Dag( int smop, Hup h) {
        Dag.smop = smop;
        this.h = h;  //4
    }
}

class Hup {
    double gep;
    Hup( double g) {
        gep = g; //2
    }
}

class TestDag {
    Dag methodB( Hup p) {
        Dag t;  //3
        t = new Dag( 2, p);
        t = new Dag( 3, p);
        return t; //5
    }
    main() {
        Dag y;  //1
        y = methodB( new Hup( 13));
        //6
    }
}
public class Stove {
    int heat = 5;
    public boolean equals(Stove s) {
        return this.heat == s.heat;
    }
    public boolean equals(Object o) {
        return this.heat == ((Stove)o).heat;
    }
    public static void main( String []args) {
        Object o = new Stove();
        Stove s1 = new Stove();
        Stove s2 = new Stove();
        System.out.println( s1.equals(o));
        System.out.println( o.equals(s1));
        System.out.println( s1.equals(s2));
    }
}

1. Will this compile?
2. Which equals method overrides Object’s?
3. Are the equals methods overloaded?
4. s1.equals( o) which method is called?
5. o.equals( s1) which method is called?
6. s1.equals( s2) which method is called?
7. Is the explicit cast an upcast or downcast?
8. Is the explicit cast safe? If not, what would safe syntax be?
What is print out?

```java
File f = new File( "myfile.txt" );
Scanner scnr = new Scanner( f );
scnr.nextLine();
int count = 4;
if ( scnr.hasNextInt())
    count = scnr.nextInt();
else {
    scnr.nextLine();
    count = scnr.nextInt();
}
for ( int i = 0; i < count; i++)
    scnr.nextLine();
System.out.println( scnr.nextLine());
```

3
hello
2
8 line
5 line
3 line
4 line
other
What is the bug?

```java
class IntList {
    private ArrayList<Integer> list = new ArrayList<>();

    public void add( int i) {
        list.add( new Integer(i));
    }

    public void add2( int i, int j) {
        add( i);
        add( j);
    }
}

class TrackCount extends IntList {
    int count = 0;

    public void add( int i) {
        this.count++;
        super.add( i);
    }

    public void add2( int i, int j) {
        this.count += 2;
        super.add2( i, j);
    }
}

//TEST CODE
TrackCount tc = new TrackCount();
tc.add(3); tc.add2( 2, 4);
System.out.println( "tc.count=" + tc.count);
```
What is the bug?

public class Super {
    public Super() {
        overrideMe();
    }
    public void overrideMe() {
    }
}

//Test Code
Sub sub = new Sub();
sub.overrideMe();

public final class Sub extends Super {
    private final Date date;
    Sub() {
        date = new Date();
    }
    @Override
    public void overrideMe() {
        System.out.println(date);
    }
}

From Joshua Bloch, Effective Java
Current Sequence:

CS302
- Structured and OO Programming

CS367
- Basic and Advanced Data Structures

New Sequence Fall 2017

CS200
- Structured Programming

CS300
- OO Programming
- Basic Data Structures

CS400
- Advanced Data Structures
- Software Project Skills
Good Luck on your Final Exams!