

Lecture 11 (review session for intro to computing)

=====

1)

(Big-Oh notation continued)

Assume that each of the following operations takes 1 time unit: an arithmetic op, a comparison, a logical op, an assignment, accessing a particular row in an array, accessing a particular column in an array, a variable declaration (allocation of space in memory for the declared variable).

```
public void fillArray(int[][] arr, int n)
{
    int row, col;

    for (row=0; row < n; row++)
        for (col=0; col < n; col++)
            arr[row][col] = 10*row + col;
}
```

- a) Find the worst-case time, $T(n)$ for the method fillArray above.
 - b) $T(n)$ is $O(?)$
- =====

2)

(2-dimensional arrays)

what are the contents of the array arr above (from the method fillArray) when $n = 5$?

=====

3)

(Big-Oh notation)

We have an algorithm which has time $T(n) = 4n + 4$,

- a) To find the worst case time, if we pick the value 5 as the constant c , for the inequality $T(n) \leq c*f(n)$ ($4n + 4 \leq 5n$ in this case) what is the value of n_0 ?
 - b) If we pick the value 6 as the constant c , for the inequality $T(n) \leq c*f(n)$ ($4n + 4 \leq 6n$ in this case) what is the value of n_0 ?
- =====

4)

(pre-increment/post-increment pre-decrement/post-decrement operators)

what are the values of result1 - result4?

```
public static void main(String[] args)
{
    int i=25, j=30, k=4, l = 9;
    int result1, result2, result3, result4;

    result1 = 5 + ++i;
    result2 = 87 / --j;
    result3 = 50 % k++;
    result4 = 6 * l--;
}
```

=====

lecture11

5)
(Primitive and class types and call/return-by-value)

Remember:

In Java you make a class cloneable by having it implement the Cloneable interface; e.g., to make class Person cloneable you would write:

```
public class Person implements Cloneable {
    // the usual class variables, constructors, and methods

    // go here
}
```

- a) what are the 3 lines that the main method in class TestNumber prints out?
- b) what are the 2 lines that the main method in class TestVector prints out?
- c) what changes would you make in class TestVector to ensure that the Vector v in that class cannot be inappropriately modified?

```
public class TestNumber {
    private int number;

    public void setNumber(int n)
    {
        number = n;
    }

    public int getNumber()
    {
        return number;
    }

    public static void main(String[] args)
    {
        TestNumber newNum = new TestNumber();
        newNum.setNumber(25);
        System.out.println("number is: " + newNum.getNumber());
        int n1 = newNum.getNumber();
        n1 = 90;
        System.out.println("number is: " + newNum.getNumber());
        System.out.println("n1 is: " + n1);
    }
}
```

```
public class TestVector {
    private Vector v;

    public void setvector(int[] arr)
    {
        v = new Vector(arr.length);
        for (int i = 0; i < arr.length; i++)
            v.addElement(new Integer(arr[i]));
    }

    public Vector getvector()
    {
        return v;
    }

    public void printvector()
    {
        Integer n;
        System.out.print("vector v has contents: ");
        for (int i=0; i < v.size(); i++) {
            n = (Integer)v.elementAt(i);
            System.out.print(" " + n.intValue());
        }
    }
}
```

```

        System.out.println(".");
    }

    public static void main(String[] args)
    {
        int[] arr1 = {1, 2, 3, 4, 5};
        TestVector newVec = new TestVector();
        newVec.setVector(arr1);
        newVec.printVector();
        Vector n1 = newVec.getVector();
        n1.addElement(new Integer(90));
        n1.addElement(new Integer(-25));
        newVec.printVector();
    }
}

```

6)

(static and non-static variables)

- a) What is the line printed out by the main method of class TestChar1?
- a) What is the line printed out by the main method of class TestChar2?

```

public class TestChar1 {
    private char c;

    public void setChar(char ch)
    {
        c = ch;
    }

    public char getChar()
    {
        return c;
    }

    public static void main(String[] args)
    {
        TestChar1 newChar1 = new TestChar1();
        newChar1.setChar('a');
        TestChar1 newChar2 = new TestChar1();
        newChar2.setChar('z');
        char c1 = newChar1.getChar();
        char c2 = newChar2.getChar();
        System.out.println("c1 is " + c1 + " c2 is " + c2);
    }
}

```

```

public class TestChar2 {
    public static char c;

    public void setChar(char ch)
    {
        c = ch;
    }

    public char getChar()
    {
        return c;
    }

    public static void main(String[] args)
    {
        TestChar1 newChar1 = new TestChar1();
        newChar1.setChar('a');
        TestChar1 newChar2 = new TestChar1();
    }
}

```

lecture11

```
newChar2.setChar('z');
char c1 = newChar1.getChar();
char c2 = newChar2.getChar();
System.out.println("c1 is " + c1 + " c2 is " + c2);
}
}
```

7)

(spot the errors -- 7 compile-time, 2 run-time)

```
import java.util.*;
public class Test{

    String s1 = "hey";
    Vector strVec = new Vector(20);
    char ch = "a";

    public void setVector(String[] strs)
    {
        int i = 0;
        do {
            strVec.addElement(strs[i]);
        } while (i < strs.length)
        strVec.trimToSize();
    }

    public static void main(String[] args)
    {
        String a = {"one", "two", "three", "four"};
        setVector(a);
        String b[] = {};
        setVector(b);
    }
}
```

8)

(iteration and recursion)

write a method `int square(int n)` that given a number `n` returns the value of `n*n` (`n` squared) without using the multiplication operator. You may use your knowledge of multiplication via addition from homework 1.

- a) Your method `int square(int n)` should call a recursive method that actually computes the square of `n`.
- b) Your method `int square(int n)` should call an iterative method that actually computes the square of `n`.

(Tracing through a recursive method)

9)

write out a trace of the return values when `fib(5)` is called.

```
public int fib(int n)
{
    // calculates fib(n) where n >= 0
    if (n == 0)
```

lecture11

```
    return 0;
else if (n == 1)
    return 1;
else
    return fib(n-1) + fib(n-2)
}
```

10)

(inheritance)

Imagine that you have defined a class called BaseClass which has the following attributes:

```
    private int num;
    private String arg;
and the following methods:
```

```
    public void doSomething(String app);
    private void appendArg(String app);
    public int getNum();
    public void setNum(int n);
    public String getArg();
    public void setArg(String a);
```

You create a new class DerivedClass that is a subclass of BaseClass and define a variable newClass in DerivedClass's main method as follows:

```
DerivedClass newClass = new DerivedClass();
```

which of the following calls in main would be invalid:

```
int a = newClass.getNum();
newClass.setNum(56);
newClass.setArg("happy happy");
newClass.appendArg("joy joy");
String newStr=newClass.getArg();
```

11)

(method overloading)

what's wrong here?

```
import java.util.*;
public class Area{

    double area(int s1, double s2)
    {
        return(s1*s2);
    }

    double area(int s1, int s2)
    {
        return(s1*s2*1.0);
    }

    double area(double r)
    {
        return(Math.PI*r*r);
    }

    int area(int s1, int s2)
    {
        return(s1*s2);
    }
}
```

}

12)

(exceptions)

what's the difference in output between a call to
 double quot = divide1(5, 0);
 and
 double quot = divide2(5, 0);

```
public double divide1(int numerator, int denominator)
{
    double quotient = SENTINEL;
    try
    {
        if (denominator == 0)
            throw new DivideByZeroException();
        quotient = numerator/(double)denominator;
        System.out.println(numerator + "/"
                           + denominator
                           + " = " + quotient);
    }
    catch(DivideByZeroException e)
    {
        System.out.println(e.getMessage());
        System.exit(0);
    }
    return(quotient);
}

public double divide2(int numerator, int denominator)
{
    double quotient = SENTINEL;
    try
    {
        if (denominator == 0)
            throw new DivideByZeroException();
        quotient = numerator/(double)denominator;
        System.out.println(numerator + "/"
                           + denominator
                           + " = " + quotient);
    }
    catch(DivideByZeroException e)
    {
        System.out.println(e.getMessage());
    }
    finally {
        return(quotient);
    }
}
```

13)

(command line input to a program)

a) If I've defined a java class TestNew and compiled it,
 what is the value of args.length in main after I
 type in the following command?:

```
java TestNew input.txt output.txt
```

b) what would the values stored in args be?

lecture11

14)

We want to copy the contents of one file to another in the class defined below. What expression should go in the parentheses after the while statement. What statements should occur in the body of the while loop?

What happens when a `FileNotFoundException` occurs in main given the class definition below (if for example, `input.txt` cannot be found)?:

```
public class CopyFile{
    public static void main(String[] args)
    {
        try {
            FileReader fr = new FileReader("input.txt");
            FileOutputStream fw = new FileOutputStream("output.txt");
            BufferedReader in = new BufferedReader(fr);
            PrintWriter out = new PrintWriter(fw);
            String getLine = in.readLine();
            while (
                ) {

                }
            in.close();
            out.close();
        } catch (Exception e2) {
            System.err.println("An exception occurred");
            return;
        } catch (IOException e1) {
            System.err.println("An IO exception occurred");
            return;
        } catch (FileNotFoundException e) {
            System.err.println("File not found!");
            return;
        }
    }
}
```