Midterm Exam 2 for CS161:
Fall 2015
Instructor: C. Anderson
100 points total

NAME: ANSWERS

Answers to counting questions must be written as mathematical expressions that may include factorial operations. Use the last page if you need more space.

1. **[10 points]** For this definition of $f(n)$,

$$f(n) = \begin{cases} 1 & n = 0 \\ 1 & n = 1 \\ 2 & n = 2 \\ f(n-1)f(n-3) & n \geq 3, \end{cases}$$

write a recursive Java method that returns the value of $f(n)$.

```
public int f( int n) {
    if (n ==0 || n ==1)
        return 1;
    if (n ==2)
        return 2;

    return f(n-1) * f(n-3);
}
```

Now write the values of $f(n)$ for $n = 0, 1, 2, 3, \ldots, 8$.

1, 1, 2, 2, 2, 4, 8, 16, 64

2. **[10 points]** Write the printed result of this code below the code.

```
public class A {

    public String silly(String s) {
        if (s.length() <= 1)
            return s;

        if (Character.isUpperCase(s.charAt(0)))  // Assume isUpperCase works.
            return s.charAt(0) + silly(s.substring(1));
        else
            return silly(s.substring(1));
    }

    public static void main(String[] args) {
        A a = new A();
        System.out.println(a.silly("AbCdE"));
        System.out.println(a.silly("Yikes"));
    }
}
```

ACE
Ys

3. **[10 points]** How many ways can the elements in a 10 element array be ordered?

10!

4. **[10 points]** How many 5-letter variable names of all lowercase letters must we have to guaranteee that at least two of them start with the same letter?

26 + 1 = 27

5. [5 points] How many committees of 3 people can be formed from the set of people {Beth, Bill, Jim, Mary, Janet, Kim}? Your answer must include factorial expressions.

$$C(6,3) = \frac{6!}{3!(6-3)!} = \frac{6!}{3!\,3!}$$

6. [5 points] How many committees of 3 people can be formed from the set of people {Beth, Bill, Jim, Mary, Janet, Kim} if the committees must include Beth? Your answer must include factorial expressions.

$$C(5,2) = \frac{5!}{2!(5-2)!} = \frac{5!}{2!\,3!}$$

7. [10 points] How many variable names are possible in a programming language that allows variable names to have only lowercase letters and can be 1, 2, 3, 4, or 5 letters long?

$$26 + 26^2 + 26^3 + 26^4 + 26^5$$

8. [10 points] What is the minimum number of processes that must be run on a cluster of 3 workstations to guarantee that at least one of the workstations is running 5 processes?

$$3 \cdot 4 + 1 = 13$$

9. [2 points] Circle **True** or **False**: A recursive method must have only one termination test.

10. [2 points] Circle **True** or **False**: In the body of a constructor for a class, you may call another version of the constructor having different arguments by typing `this` followed by argument values within parentheses.

11. [2 points] Circle **True** or **False**: Counting the number of possible license plates involves permutations.

12. [2 points] Circle **True** or **False**: Counting the number of hands containing 5 cards that can be dealt from a deck of playing cards involves 5-combinations.

13. [2 points] Circle **True** or **False**: $C(n,r) = r!\,P(n,r)$.

14. [**10 points**] To the right of each print statement, write what is printed when the `main` method in `Test.java` is run?

```
========================== X.java ==========================
public class X {
    protected String me;
    public X(String s) {
        me = s;
    }
    public String getMe() {
        return me + " I am an X" + " " + getYou();
    }
    public String getYou() {
        return "X is nuts.";
    }
}
========================== Y.java ==========================
public class Y extends X {
    public Y(String s) {
        super(s);
    }
    public String getYou() {
            return "Y is nuts";
    }
}
========================== Z.java ==========================
public class Z extends Y {
    public Z(String s) {
        super(s);
    }
}
========================== Test.java ==========================
public class Test {
    public static void main(String[] args) {
        X x = new X("xx");
        Y y = new Y("yy");
        Z z = new Z("zz");
        System.out.println( x.getMe() );
        System.out.println( y.getMe() );
        System.out.println( z.getMe() );
        X humm = new Z("humm");
        System.out.println( humm.getMe() );
        Y what = new Z("what");
        System.out.println( what.getMe() );
    }
}
```

xx I am an X X is nuts.

yy I am an X Y is nuts.

zz I am an X Y is nuts.

humm I am an X Y is nuts.

what I am an X Y is nuts.

15. **[10 points]** Prove by induction that for every positive integer n,

$$1 \cdot 2^1 + 2 \cdot 2^2 + 3 \cdot 2^3 + \cdots + n \cdot 2^n = (n-1)\,2^{n+1} + 2$$

State and label the proof of the base case, the inductive hypothesis, and the proof of the inductive step. Clearly indicate where you have used the inductive hypothesis.

Base Case: $P(n=1)$ $\quad 1 \cdot 2^1 \stackrel{?}{=} (1-1)\,2^{1+1} + 2 = 2$

$2 = 2$ True

Inductive Hypothesis. Assume $P(n=k)$ is True: $1 \cdot 2^1 + 2 \cdot 2^2 + \cdots + k2^k = (k-1)\,2^{k+1} + 2$.

Inductive Step. Prove $P(n=k) \rightarrow P(n=k+1)$

Definition of $P(n=k+1)$

$$\underbrace{1 \cdot 2^1 + 2 \cdot 2^2 + \cdots + k2^k} + (k+1)\,2^{k+1} \stackrel{?}{=} (k+1-1)\,2^{k+1+1} + 2$$

By Ind. Hyp.: $(k-1)\,2^{k+1} + 2 + (k+1)\,2^{k+1} \stackrel{?}{=} k\,2^{k+2} + 2$

Algebra:

$$2^{k+1}(k-1+k+1) \stackrel{?}{=} k\,2^{k+2}$$

$$2^{k+1} \cdot 2k \stackrel{?}{=} k\,2^{k+2}$$

$$k\,2^{k+2} = k\,2^{k+2}$$

True