

CSE 142, Spring 2009
Midterm Exam – Answer Key

1. Expressions

For each expression in the left-hand column, indicate its value in the right-hand column.

Be sure to list a constant of appropriate type and capitalization.

e.g., 7 for an int, 7.0 for a double, "hello" for a String, true or false for a boolean.

<u>Expression</u>	<u>Value</u>
$3 * (7 + 1) - 3 - 7 + 2$	16
$9 \% 5 + 20 \% 5 + 20 \% 3$	6
$2.3 * 2.0 + 1.5 * 10 / 3$	9.6
$1 + 2 + "3 + 4" + 5 + 6 / 4$	33 + 451
$5 \% 2 + 10 / 5 / 2 + 4.2$	6.2

2. Parameter Mystery

What output is produced by the following program?

```
public class ParameterMystery {
    public static void main(String[] args) {
        String head = "shoulders";
        String knees = "toes";
        String elbow = "head";
        String eye = "eyes and ears";
        String ear = "eye";

        touch(ear, elbow);
        touch(elbow, ear);
        touch(head, "elbow");
        touch(eye, eye);
        touch(knees, "Toes");
        touch(head, "knees " + knees);
    }

    public static void touch(String elbow, String ear) {
        System.out.println("touch your " + elbow + " to your " + ear);
    }
}
```

```
touch your eye to your head
touch your head to your eye
touch your shoulders to your elbow
touch your eyes and ears to your eyes and ears
touch your toes to your Toes
touch your shoulders to your knees toes
```

3. If/Else Simulation

For each call below to the following method, write the output that is produced, as it would appear on the console:

```
public static void ifElseMystery(int x, int y) {
    int z = 4;
    if (z <= x) {
        z = x + 1;
    } else {
        z = z + 9;
    }
    if (z <= y) {
        y++;
    }

    System.out.println(z + " " + y);
}
```

<u>Method Call</u>	<u>Output</u>
<code>ifElseMystery(3, 20);</code>	13 21
<code>ifElseMystery(4, 5);</code>	5 6
<code>ifElseMystery(5, 5);</code>	6 5
<code>ifElseMystery(6, 10);</code>	7 11

4. While Loop Simulation

For each call below to the following method, write the output that is produced, as it would appear on the console:

```
public static void whileMystery(int n) {
    int x = 0;
    int y = 1;
    while(n > x && n > y) {
        n--;
        x = x + 2;
        y = y + x;
    }
    System.out.println(x);
}
```

<u>Method Call</u>	<u>Output</u>
<code>whileMystery(5);</code>	4
<code>whileMystery(10);</code>	6
<code>whileMystery(4);</code>	2

5. Assertions

For each of the five points labeled by comments, identify each of the assertions in the table below as either being *always* true, *never* true, or *sometimes* true / sometimes false.

```
public static int assertionMystery(Scanner console) {
    int a = -1;
    int b = 0;
    int next = console.nextInt();
    // Point A
    while(next != 0) {
        // Point B
        b = Math.max(a, next);
        a = next;
        // Point C
        if(a == 42) {
            // Point D
            a++;
        }
        next = console.nextInt();
    }
    // Point E
    return b;
}
```

Fill in each box below with one of ALWAYS, NEVER or SOMETIMES. (You may abbreviate them as A, N, or S.)

	next == 0	a > 0	b >= next
Point A	SOMETIMES	NEVER	SOMETIMES
Point B	NEVER	SOMETIMES	SOMETIMES
Point C	NEVER	SOMETIMES	ALWAYS
Point D	NEVER	ALWAYS	ALWAYS
Point E	ALWAYS	SOMETIMES	SOMETIMES

6. Programming

Write a program that simulates an ant trying to crawl up a building of height 6 steps. The ant starts on the ground, at height 0. Each iteration, the ant either crawls up one step, or slips off and falls all the way back to the ground. There is a 50% chance on each iteration that the ant will slip. The program should keep going until the ant gets to the top of the building. It should then print out the number of falls that the ant took before it finally reached the top.

Here is a sample execution:

```
number of falls: 8
```

```
// this program includes the class declaration etc so that it can run - this isn't
// needed for a correct answer
import java.util.*;

public class Ant {
    public static void main(String[] args) {
        Random rand = new Random();
        int height = 0;
        int falls = 0;
        while (height<6) {
            int r = rand.nextInt(2);
            // 50% chance the ant will fall ....
            if (r==0) {
                // ant fell
                height = 0;
                falls++;
            } else {
                // ant climbs a step
                height++;
            }
        }
        System.out.println("number of falls: " + falls);
    }
}
```

Note: this solution assumes that if the ant slips even when `height==0` that counts as a fall. (Maybe the ant gets partway up the wall?) We also counted as correct solutions that didn't count this case as a fall.

7. Programming

Write a static method `isPowerOfTwo` that takes an integer `n` as an argument, and that returns `true` if `n` is a power of two, and otherwise `false`. If `n` is zero or negative, return `false`. Note that `isPowerOfTwo(1)` should return `true`, since $2^0=1$.

Here are some example calls to the method and the value it returns:

Call	<code>isPowerOfTwo(0)</code>	<code>isPowerOfTwo(1)</code>	<code>isPowerOfTwo(5)</code>	<code>isPowerOfTwo(8)</code>
Result	false	true	false	true

```
// this program includes the class declaration etc so that it can run - this isn't
// needed for a correct answer
import java.util.*;
```

```
public class PowerOfTwo {
    // main not required for the answer on the test
    public static void main(String[] args) {
        for (int i=0; i<100; i++) {
            if (isPowerOfTwo(i)) {
                System.out.println(i);
            }
            if (isPowerOfTwoA(i)) {
                System.out.println(i);
            }
        }
    }

    // a couple of possible solutions
    public static boolean isPowerOfTwo (int n) {
        int reduced = n;
        if (n<1) {
            return false;
        }
        while (reduced>1) {
            if (reduced%2 == 0) {
                reduced = reduced/2;
            } else {
                return false;
            }
        }
        return true;
    }

    public static boolean isPowerOfTwoA (int n) {
        int pow = 1;
        while (pow<=n) {
            if (n==pow) {
                return true;
            }
            pow = 2*pow;
        }
        return false;
    }
}
```

8. Programming

A “perfect number” is a positive integer that is the sum of all its proper factors (that is, factors including 1 but not the number itself). The first two perfect numbers are 6 and 28, since $1+2+3=6$ and $1+2+4+7+14=28$. Write a static method `findAllPerfectUpto` that takes an integer `max` as an argument and prints out all perfect numbers that are less than or equal to `max`.

Here is the console output from a call to `findAllPerfectUpto(6)`:

```
Perfect numbers up to 6: 6
```

Here is the console output from a call to `findAllPerfectUpto(500)`:

```
Perfect numbers up to 500: 6 28 496
```

```
// this program includes the class declaration etc so that it can run - this isn't
// needed for a correct answer
import java.util.*;
public class Perfect {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        int n = 1; // any number other than 0 to start the loop
        while (n!=0) {
            System.out.print("Enter an integer: ");
            n = console.nextInt();
            findAllPerfectUpto(n);
        }
        System.out.println("bye");
    }

    public static void findAllPerfectUpto(int n) {
        System.out.print("Perfect numbers up to " + n + ":");

        for (int current = 1; current <= n; current++) {
            int sum = 0;
            //
            for (int factor = 1; factor < current; factor++) {
                if (current % factor==0) {
                    sum = sum+factor;
                }
            }

            if (sum == current) {
                System.out.print(" " + current);
            }
        }
        System.out.println();
    }
}
```

X. Extra Credit (+1 point)

Make up an English word for a monster that is part Alan, part Dan. Use only the letters A, L, D, and N. Example: Landalad. Do not use this example.

(Any other word you write will get the +1 extra point.)