

COS 126	General Computer Science	Fall 2003
<b>Exam 1</b>		

This test has 10 questions worth a total of 50 points. You have 120 minutes. The exam is closed book, except that you are allowed to use a one page cheatsheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. **Write out and sign the Honor Code pledge before turning in the test.**

*“I pledge my honor that I have not violated the Honor Code during this examination.”*

Problem	Score
0	
1	
2	
3	
4	
Sub 1	

Problem	Score
5	
6	
7	
8	
9	
Sub 2	

Total	
-------	--

**Name:**

**Login ID:**

**Precept:**

1	MF 10:00	Paul
2	MF 11:00	Donna
3	MF 1:30	Shirley
4	MF 2:30	Kevin
5	M 7:30	Kevin
	F 2:30	



**2. Data types. (4 points)**

What does the following program print out? Circle your answer.

```
public class JavaBasics {
    public static void main(String[] args) {

        double a = Math.sqrt(330 / 70);
        System.out.println(a);

        boolean b = (((13 > 0) || (4 < 3)) && (21 % 7 == 0));
        System.out.println(b);

        int c;
        for (c = 0; c < 10; c++)
            c = c + 3;
        System.out.println(c);

        String d0 = "";
        String d1 = "1" + d0 + "0" + d0 + "1";
        String d2 = "0" + d1 + "1" + d1 + "0";
        System.out.println(d2);
    }
}
```

### 3. Debugging. (5 points)

For each of the following programs, indicate which error (from the list on the facing page) you will receive when you try to compile and run the following programs. Write down the letter in the space provided.

```
----- public class Bugs {
        public static void main(String[] args) {
            a = 5;
        }
    }

----- public class Bugs {
        public static void main(String[] args) {
            int a = Double.parseDouble(args[0]);
        }
    }

----- public class Bugs {
        public static void recur(int i) {
            recur(i-1);
        }
        public static void main(String[] args) {
            int n = Integer.parseInt(args[0]);
            recur(n);
        }
    }

----- public class Bugs {
        public static String f() {
            String s = "";
            for(int i = 0; i < 3; i++) {
                s = s + "*";
            }
        }
        public static void main(String[] args) {
            String result = f();
        }
    }

----- public class Bugs {
        public static void main(String[] args) {
            int a = 4
            int b = 5;
            int c = 6;
            System.out.println(c);
        }
    }
```

- (a) Bugs.java:n: ';' expected
- (b) Bugs.java:n: cannot resolve symbol
- (c) Bugs.java:n: missing method body, or declare abstract
- (d) Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0
- (e) Exception in thread "main" java.lang.NumberFormatException
- (f) java.lang.StackOverflowError
- (g) Bugs.java:n: possible loss of precision
- (h) Bugs.java:n: ')' expected
- (i) Bugs.java:n: missing return statement

**4. Loops and conditionals. (4 points)**

Consider the following program.

```
public class Pattern {  
  
    public static void main(String[] args) {  
        int N = Integer.parseInt(args[0]);  
  
        for (int j = -N; j <= N; j++) {  
            for (int i = -N; i <= N; i++) {  
                if (i == j) System.out.print("A ");  
                else if (i == -j) System.out.print("B ");  
                else System.out.print(". ");  
            }  
            System.out.println();  
        }  
    }  
}
```

(a) What is the result of the following command?

```
java Pattern 1
```

(b) What is the result of the following command?

```
java Pattern 2
```

## 5. Input, output, loops, arrays, debugging. (6 points)

```
public class Lyrics {
    public static void main(String args[]) {
        int N = Integer.parseInt(args[0]);
        String[] s = new String[N];
        for (int i = 0; i < N; i++)
            s[i] = StdIn.readString();
        int x = 2;
        while (x != 0) {
            System.out.println(s[x]);
            x = (x + 3) % N;
        }
    }
}
```

What happens when you execute the program above with the following commands? Circle your answers. Assume that the file `aesoprock.txt` contains the following text.

```
It goes thieves bandits lowlives scum
Punks that buckle under the rumble of my drum
```

(a) `java Lyrics 10 < aesoprock.txt`

(b) `java Lyrics 6 < aesoprock.txt`

(c) `java Lyrics 10 < aesoprock.txt | java Lyrics 2`

**6. Using arrays. (4 points)**

When possible, it is usually best to avoid using arrays. Circle each of the following situations which can be done *without* using an array (or storing all of the data).

- (a) Read in  $N$  integers and print their average.
- (b) Read in  $N$  integers and print out those outliers that are 25% or more above the average.
- (c) Read in  $N$  integers and print them back out in ascending order.
- (d) Read in  $N$  integers and determine whether they are in ascending order.
- (e) Generate  $N$  random e-mail addresses (not necessarily valid) and print them out. Duplicates are allowed.
- (f) Read in a list of  $N$  CD tracks and generate a random playlist of  $N$  songs such that each song is played exactly once.

**7. Functions. (5 points)**

Write a function `min6` that takes six integer inputs and returns the minimum value. You are encouraged to use the function `min3`, which takes three integer inputs and returns the minimum value. You will lose one point for each `if` statement that you write. Put your answer in the box below.

```
static int min3(int a, int b, int c) {
    if (a < b && a < c) return a;
    if (b < c) return b;
    return c;
}

static int min6(int a, int b, int c, int d, int e, int f) {

}
```

## 8. Recursion. (6 points)

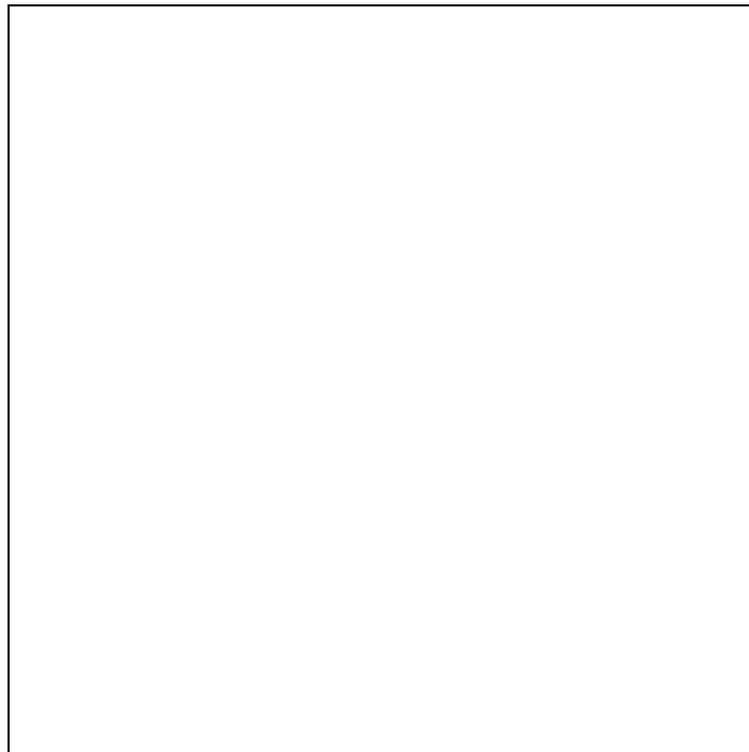
Suppose that the function `drawCircle(x, y, s)` plots a circle centered at  $(x, y)$  of diameter  $s$ . Consider the following recursive function.

```
static void recur(double x, double y, int n, double s) {
    if (n <= 0) return;

    recur(x - s/2, y + s/2, n - 1, s/2);
    drawCircle(x, y, s);
    Turtle.pause(10);
    recur(x + s/2, y + s/2, n - 1, s/2);
}
```

- (a) What does `recur(256, 256, 3, 256)` plot? Your drawing and geometry does not have to be perfect to get full credit (but your recursion should be).

(512, 512)



(0, 0)

- (b) Label the circles in your figure above in the order that they get plotted by writing 0th, 1st, 2nd, and so forth in the center of the corresponding circle.

**9. TOY. (5 points)**

Suppose that you load the following into locations 10–17 of TOY, set the PC to 10, and press RUN.

00: AAAA  
01: BBBB  
02: 0000

10: 8A00  
11: 8B01  
12: 8C02  
13: CC16  
14: 9A01  
15: C017  
16: 9B00  
17: 0000

(a) Upon termination, what is stored in memory locations 00, 01, and 02? Circle your answers.

(b) Repeat the above question, but assume memory locations 00, 01, and 02 initially contain AAAA, BBBB, and 0005, respectively.

10. **TOY. (5 points)**

Suppose that you load the following into locations 10–16 of TOY, set the PC to 10, and press RUN. *Hint:* recall that the XOR of two bits is 1 if they are different and 0 if they are the same. The bitwise XOR of two 16-bit integers is the result of independently applying the XOR function to each of the 16 pairs of corresponding bits.

```

10: 7C00   RC <- 0000
11: 8AFF   read RA
12: CA15   if (RA == 0) pc <- 15
13: 4CCA   RC <- RC ^ RA
14: C011   pc <- 11
15: 9CFF   write RC
16: 0000   halt

```

- (a) What hexadecimal integer is printed to standard output if the following integers appear on standard input? Circle your answer.

1234 1234 5678 5678 0000

- (b) What hexadecimal integer is printed to standard output if the following integers appear on standard input? Circle your answer. *Hint: use your insight from part (a) and do not do any tedious calculations.*

DEAD BEEF CAFE FACE BODE D1CE FADE BEAD FEED CEDE CODE FOOD ACDC  
 DEAD BEEF FACE CAFE BODE D1CE FADE BEAD FEED CEDE CODE FOOD 0000

## TOY REFERENCE CARD

## INSTRUCTION FORMATS

	. . . . .	. . . . .	. . . . .	. . . . .	
Format 1:	opcode	d	s	t	(0-6, A-B)
Format 2:	opcode	d	addr		(7-9, C-F)

## ARITHMETIC and LOGICAL operations

1: add	R[d] <- R[s] + R[t]
2: subtract	R[d] <- R[s] - R[t]
3: and	R[d] <- R[s] & R[t]
4: xor	R[d] <- R[s] ^ R[t]
5: shift left	R[d] <- R[s] << R[t]
6: shift right	R[d] <- R[s] >> R[t]

## TRANSFER between registers and memory

7: load address	R[d] <- addr
8: load	R[d] <- mem[addr]
9: store	mem[addr] <- R[d]
A: load indirect	R[d] <- mem[R[t]]
B: store indirect	mem[R[t]] <- R[d]

## CONTROL

0: halt	halt
C: branch zero	if (R[d] == 0) pc <- addr
D: branch positive	if (R[d] > 0) pc <- addr
E: jump register	pc <- R[d]
F: jump and link	R[d] <- pc; pc <- addr

Register 0 always reads 0.

Loads from mem[FF] come from stdin.

Stores to mem[FF] go to stdout.