Computer Science 241
Test 1
Feb 16, 2005

1. Use the terms on the last page to fill in the blanks to describe the program also on the last page (terms may be used more than once). (2 points each.)

The names `a` and `f` are examples of _____`identifier`_____s . `a` is the name of a(n) _____`variable`_____ and `f` is the name of a(n) _____`method`_____ . On line 1, we find the _____`declaration`_____ of `a`, and the on the following line is its _____`initialization`_____ . The word `int` specifies the _____`type`_____ of `a`. As opposed to `a`, 5 is a(n) _____`literal`_____ .

On line 3, "`a++`" is a(n) _____`expression`_____, as opposed to "`a++;`" which is a(n) _____`statement`_____ . One line 4, `"a: " + a` creates a new String by _____`concatenation`_____ . On line 5, `f(a)` is a(n) _____`invocation`_____ of `f`. Notice that `f(a)` has an `int` value, even though `b` should contain a double value. A(n) _____`automatic cast`_____ makes this work correctly.

Even though `a` on line 1 and `a` on line 9 have the same name, they are different because they each have a different _____`scope`_____ . Even though `f` on line 8 and `f` on line 16 have the same name, they are different because they each have a different _____`signatures`_____ , namely `f(int)` and `f(int, int)`, respectively. That these two things have the same name is called _____`overloading`_____ .

The loop in lines 10-13 is a(n) _____zero-trip_____ loop, whereas the loop in lines 17-22 is a(n) _test-in-the-middle_ loop.

2. The output of running this piece of code is (5 points):

```
a: 6
3.0
```

3. For each pice of code, show the value of `a` and `i` at the beginning of each iteration of the loop and after the loop finishes. (6 points each)

```
int a = 7;
int i = 0;

do {
   a /= 2;
   i++;
} while (a > 1);
```

iteration	1	2	end
a	7	3	1
i	0	1	2

```
int a = 7;
int i = 0;

for(;;) {
   a /= 2;
   if (a <= 1) break;
   i++;
}
```

iteration	1	2	end
a	7	3	1
i	0	1	1

4. We have seen several versions of programs that average a series of numbers supplied by the user. Write an algorithm that allows a user to input a series of integers (using -1 as a sentinel value to signal being finished) and computes the *range* of the values, that is, the difference between the largest and smallest. You may assume the first input is not -1. (10 points.)

```
- Input first number from user, store in query
- Set smallest = query
- Set largest = query
- Loop
      - Input next number from user, store in query
      - If query == -1, break
      - If query < smallest, set smallest = query
      - If query > largest, set largest = query
- Set range = largest - smallest
- Display range
```

5. Write *two* methods, one *iterative* and one *recursive*, to compute the sum of the first n positive integers, that is $1 + 2 + \ldots + n$. (If you know the explicit formula for the sum of an arithmetic sequence, do not use it.) (8 points each.)

Iterative version:

```
static int sum (int n) {
    int s = 0;
    for (int i = 1; i <= n; i++)
        s += i;
    return s;
}
```

Recursive version:

```
static int sum (int n) {
    if (n == 1)
        return 1;
    else
        return n + sum(n-1);
}
```

6. You have used the standard Java method *str*`.substring(int, int)` that returns a portion of a string bounded by the two given integers. Suppose that Java did not provide such a method. Instead, write your own method using *str*`.charAt(int)`. In other words, write a body for the following method which accepts a String and two ints, indicating an (inclusive) starting position and (exclusive) ending position and returns an appropriate String. (Do not worry about checking for correct arguments; that is, assume $0 \leq$ `start` $\leq$ `end` $\leq$ `str.length()`.) (12 points.)

```
static String homemadeSubstring(String str, int start, int end) {


     String toReturn = "";       // empty string
     for (int i = start; i < end; i++)
          toReturn += str.charAt(i);
     return toReturn;
}
```

7. Write a method which receives two integers, `height` and `width`, and draws a box, using asterisks, that has those dimensions. For example, it would produce the following box if `height` was 5 and `width` was:

```
****
*  *
*  *
*  *
****
```

You may assume `height` and `width` are both at least 2. (10 points.)

```
static void printBox(int height, int width) {

     // Make top/bottom line
     String top = "";
     for (int i = 0; i < width; i++)
          top += "*";

     // Make line for middle part
     String middle =  "*";
     for (int i = 0; i < width - 2; i++)
          middle +=  " ";
     middle +=  "*";

     System.out.println(top);
     for (int i = 0; i < height - 2; i++)
          System.out.println(middle);
     System.out.println(top);
}
```