

CS305j Fall 2008

Midterm 1 Key, Suggested Solutions, and Grading Criteria

Abbreviations:

NAP - no answer provided

ECF - error carried forward

OBOE - off by one error

BOD - benefit of the doubt

GCE - misunderstood question. Answer is way off base.

NN - Not Necessary. The code was not required. Usually no points off for this.

Question 1:

A. 8 B. 5.5 C. 3 D. "3UT" E. 7 F. "CS37" G. 5.5

H. 4 I. -20

2 points per answer.

missing quotes was -1 for first instance

Some error carried forward was given if a student showed their work and it was clear where the mistake was.

$5 \% 12 = 5$, not 0. Lots of mistakes on that one

Most mistakes were due to errors in order of operations or not realizing if an operation was floating point or int division.

Question 2:

10

3 4

2 5

3 2

3 2

1 8

1

2 points each

Students did very well on this tracing question for the most part. Most parts were independent so there was not the chance for error carried forward.

Question 3:

loop(-1): 0
loop(1): 2
loop(2): 6
loop(4): 20

2 points each. There were some math errors on this problem and some students thought if sent -1 the method wouldn't return anything. If sent -1 the method returns 0.

Question 4: The program drew 6 lines total, 3 vertical and 3 horizontal. The lines made the drawing panel look a 4 by 4 grid. There would be a large circle the size of the whole drawing panel and a smaller circle as large as the middle 4 squares. A lot of answers thought the smaller circle would only be as big as one of the small grid cells and positioned in the 2nd row, 2nd column. The circle is bigger than that.

Break down of points.

some lines -> 2 points
6 lines total -> 2 points
3 lines horizontal and 3 lines vertical -> 2 points
lines correctly positions and spaced -> 2 points
circles correct 2 points

Question 5:

Suggested solution:

```
public static double getArea(double x1, double y1, double x2, double y2){  
    return (x2 - x1) * (y1 - y2);  
}
```

calculate width, attempt -> 1 point
calculate width, correct -> 2 points
calculate height, attempt -> 1 point
calculate height, attempt -> 2 points
calculate area, attempt -> 1 point
calculate area, attempt -> 2 points
return value -> 1 point

Most people did well on this question. Biggest problems were not getting the

order of the operands right. $(y_2 - y_1)$ instead of $(y_1 - y_2)$

Question 6:

Suggested solution:

```
public static void showMoney(int knuts){
    System.out.print(knuts + " Knuts is ");
    System.out.print(knuts / 493 + " G, ");
    knuts = knuts % 493;
    System.out.print(knuts / 29 + " S, ");
    knuts = knuts % 29;
    System.out.println(knuts + " K");
}
```

First part of output (x Knuts is ...) -> 1 point

Calculate and print number of Galleons, attempt -> 2 points, correct -> 2 pts

Calculate and print number of Sickles, attempt -> 2 points, correct -> 3 pts

Calculate and print number of Knuts, attempt -> 2 points, correct -> 3 pts

Biggest problems were not being able to calculate number of each coin. Loops were not necessary, but some answers used them successfully.

Question 7:

Suggested Solution:

```
public static void showTerms(int num){
    System.out.print(1);
    double total = 1.0;
    for(int i = 2; i <= num; i++){
        System.out.print( " + 1/" + i);
        total += 1.0 / i;
    }
    System.out.println(" = " + total);
}
```

print out the first 1 -> 1 point

loop to do the rest of terms, attempt -> 4 points

loop to do the rest of terms, correct -> 5 points

variable to accumulate total -> 1 point

accumulate terms, attempt -> 4 points

accumulate terms, correct -> 5 points (-2 if not floating point division)

print out terms, attempt -> 2 points

print out terms, correct -> 2 points

print out total -> 1 point

Some answers used a nested loop. That is not necessary. Lots of answers did integer division instead of floating point division.