Midterm 1

CS 1323, Spring 2016, Section 10

Name (printed legibly): _____

Student number: _____

Integrity Pledge

On my honor, I affirm that I have neither given nor received inappropriate aid in the completion of this exercise.

Signed: _____

Answer all programming questions in Java.

Unless otherwise indicated, each part of a problem is worth the same number of points.

Show your work to receive partial credit.

Pay careful attention to whether the question asks for a code fragment or a complete program. Do not write a whole program when you are asked for only a few lines of code.

Also pay attention to whether you need to get input from a user or not. When you do not need to get input, the problem will say something like "you may assume that the variables' value was set somewhere else".

Pay careful attention to distinctions like int versus double and String versus char.

Try to move through short problems quickly to leave you sufficient time to write programs and code fragments.

You will have fifty minutes to take the examination.

You do not need to use import statements or exceptions on any code.

You may abbreviate System.out.println as S.o.p., and may abbreviate prompts.

1. (10 points; 2 points each)

What type of data (int, double, String, char, or boolean) would you use to store each of the following things? Do not assume that each type is used exactly once.

a) The name of the largest national park in the United States.

String

b) The average number of visitors to this national park in February, stored with two decimal places.

double

c) The number of visitors to the park in 2015.

int

d) Whether the park is open to the public on January 1 or not.

boolean

e) The year that the park was founded.

int (could also be String)

2. (10 points; 2 points each) Give the value computed for each expression below. Pay careful attention to type, especially char versus String and int versus double. If the expression is not legal, say so.

a) 47 / 5.0

9.4

b) 47 % 5

2

c) 47 / 5

9

d) 5 != 9

true

e) “true” + “false”

“truefalse”

3. (20 points; 4 points each part) Find the value assigned by each statement below. Show all intermediate steps to get partial credit. Each part is independent, with the values for any variables starting with the ones given below (do not use the results of a) in b), for example). If the expression is not legal in Java, say so.

Be sure to distinguish double and int values by **giving double values a decimal point**, even if it is a zero.

int sets = 6;

int reps = 27;

double percent = 30.3;

a) int leftovers = reps / 3 + reps;

= 27 / 3 + 27

= 9 + 27

= 36

b) int leftovers = reps / sets % 3 * 10;

= 27 / 6 % 3 * 10

= 4 % 3 * 10

= 1 * 10

= 10

c) double leftovers = percent * (2 / sets);

= 30.3 * (2 / 6)

= 30.3 * 0

= 30.0 * 0.0

= 0.0

d) double leftovers = (int) percent + 7.5;

= (int) 30.3 + 7.5

= 30 + 7.5

= 30.0 + 7.5

= 37.5

e) int leftovers = reps / sets + 2.0;

= 27 / 6 + 2.0

4 + 2.0

6.0

Cannot assign to an int--illegal

4. (15 points) You were sent to purchase a single rose for everyone at your office from Fred's Club (a warehouse store) to celebrate Valentine's Day. Your boss wants you to do this as cheaply as possible. Roses are relatively expensive if purchased individually, but cost less when purchased by the dozen.

Write a code fragment that calculates the number of dozen and individual roses to purchase to get the minimum cost. Use the variables below.

Example: Suppose your office had 28 people and roses cost $1 each or $10 per dozen. You would purchase 2 dozen roses and 4 single roses because three dozen roses would be more expensive.

```
double cost12Roses; // value given elsewhere
double cost1Rose; // value given elsewhere
// A dozen roses always costs less than purchasing 12 individual roses

int numberEmployees; // value given elsewhere
int dozenRoses;  // the number of packages of 12 flowers to purchase
int individualRoses; // the number of individual flowers to purchase

dozenRoses = numberEmployees / 12;

individualRoses = numberEmployees – dozenRoses * 12; // could also use %

if (cost1Rose * individualRoses > cost12Roses)

{

        individualRoses = 0;

        dozenRoses = dozenRoses + 1;

}
```

5. (15 points; 5 points each part) ***Trace the code fragments*** below in the tables at the right. If there is an infinite loop, trace three iterations and write "infinite loop" in the table.

a)

```
int value = 2;
if (value < 3)
    value = 7;
else if (value ==7)
    value = 12;
else if (value == 12)
    value = 14;
else
    value = 16;
```

value
2
7

b)

```
int q = 0;
int value = 30;
int d = 6;
while ( value >= d)
{
    value = value – d;
    q = q + 1;
}
```

value	q
30	0
24	1
18	2
12	3
6	4
0	5

c)

```
int value = 9;
int sum = 2;
while (value != 12)
{
    value = value + sum;
    sum = sum + 2;
}
```

value	sum
9	2
11	4
15	6
21	8
Infinite	Loop

6. (30 points) Write a complete program that sells tickets for an OU Women's basketball game. There are two kinds of seats available: upper level and lower level. Your program should ask the user how many and which kind of seat they want. If there are sufficient seats still available, the program should record the sale. When the program exits, it should print out the amount of money collected and the number of each kind of seat remaining to be sold.

You do not need to use constants or comments in your programs, although both would be welcome. As always, you may assume that users never make mistakes entering data.

Seat Type	Seats available	Cost of Ticket
Upper level	9,000	$5
Lower level	2,000	$15

A sample run of the program is below (text in bold italics is user data entry).

```
Enter the number and type of seat (upper level or lower level) or -1 to exit
10 upper level
Enter the number and type of seat (upper level or lower level) or -1 to exit
20 lower level
Enter the number and type of seat (upper level or lower level) or -1 to exit
10 upper level
Enter the number and type of seat (upper level or lower level) or -1 to exit
-1
There are 8980 upper level seats remaining
There are 1980 lower level seats remaining
We collected 400.0 today.
```

Start code on the next page, please. You may tear this page out of your examination if you wish to have it available while you're writing code. You must turn in the page with your examination.

```
import java.util.Scanner;


public class SellTickets
{

	public static void main(String[] args)
	{
		Scanner input = new Scanner(System.in);
		final int LOWER_LEVEL_SEATS = 2000;
		final int UPPER_LEVEL_SEATS = 9000;
		final double UPPER_LEVEL_COST = 5.0;
		final double LOWER_LEVEL_COST = 15.0;
```

```
            int lower = LOWER_LEVEL_SEATS;
            int upper = UPPER_LEVEL_SEATS;
            double moneyCollected=0.0;

            //Priming read
            System.out.println("Enter the number and type of seat "
                        + "(upper level or lower level) or -1 to exit");
            int seats = input.nextInt();
            String seatType = input.nextLine();
            seatType = seatType.trim(); // remove extra space after int

            while (seats != -1)
            {
                  if (seatType.equalsIgnoreCase("upper level"))
                  {
                        if (seats < upper)
                        {
                              upper = upper - seats;
                              moneyCollected = moneyCollected + seats  +
UPPER_LEVEL_COST;
                        }
                        else
                        {
                              System.out.println("We don't have enough upper
level seats left.");
                        }
                  }
                  else
                  {
                        if (seats < lower)
                        {
                              lower = lower - seats;
                              moneyCollected = moneyCollected + seats *
LOWER_LEVEL_COST;
                        }
                        else
                        {
                              System.out.println("We don't have enough lower
level seats left.");
                        }
                  }


                  // priming read
                  System.out.println("Enter the number and type of seat "
                              + "(upper level or lower level) or -1 to
exit");
                  seats = input.nextInt();
                  seatType = input.nextLine();
                  seatType = seatType.trim(); // remove extra space
            }

            System.out.println("There are " + upper + " upper level seats
remaining");
            System.out.println("There are " + lower + " lower level seats
remaining");
```

```
            System.out.println("We collected " + moneyCollected + " today.");
        }

}
```