

Assignment 1 (Review Assignment)

30 points

Due date: Thursday, Jan. 18 (before class)

The purpose of this assignment is to help you review basics of Java programming.

1. In the United States, we commonly express dates by listing the month, then the day then the year, all separated by slash marks. In Europe, however, dates are commonly expressed by listing the day first, then the month, then the year, all separated by periods.

Write a JAVA application program that will get a line of text containing a date in US form from the user, and then change this date to European form.

Begin by asking the user to enter a date in the form of month/day/year. Store this date in a String variable.

Next, use the appropriate String methods to swap the month and day parts of the date, and replace the slash marks with periods. Print the revised String to the screen.

The following is an example of what you might see on the screen when your program runs. The exact output depends on the values that the user types in while the program runs. The user's values are shown below in italics:

```
Enter a date in the form mon/day/year:  
06/17/2008
```

```
Your date in European form is:  
17.06.2008
```

Here is another example run:

```
Enter a date in the form mon/day/year:  
5/3/09
```

```
Your date in European form is:  
3.5.09
```

2. Write a java application program that will get an integer from the keyboard. Your program should then break up the integer into individual digits (using some simple math) and print them to the screen in reverse order, with each digit appearing on a separate line.

For example:

```
Please enter an integer: 57321
```

```
Your number printed in reverse order is:
```

```
1
2
3
7
5
```

3. Write a program that gets a list of grades from the user and counts the number of grades above the average grade.

Begin your program by asking the user to enter the number of grades. Store this value in a variable called `numOfGrades`. If the number is a negative number or zero, your program should print some message and stop.

Next, create an array of size `numOfGrades`. Then you will read `numOfGrades` values representing student grades from the keyboard and place them into the array.

Next, find the average grade of all the values entered, and print this value to the screen.

Finally, count the grades that are above the average grade. Display each individual grade that is above the average grade, and display the count of such grades.

For example:

```
Please enter the number of grades: 4
```

```
Enter a grade: 80
Enter a grade: 65
Enter a grade: 78
Enter a grade: 90
```

```
The average grade is: 78.25
```

```
The grades above the average grade are:
```

```
80
90
for a total of 2 grades
```

Technical Notes

- At the top of EVERY Java file you create, you must include a comment which states your name.
- Please include proper comment for your programs.
- Programs that do not compile will receive an automatic grade of "F".

Submission instructions:

To submit your programs, you need to submit your programs **electronically on Blackboard**.

The detailed steps are as follows: go to the course Assignment page on Blackboard, click Assignment 1, and then scroll down and click “Browse my computer” to upload your program files **as one zip file**. **Please also bring a hard copy of your programs to the class to submit.**