

Assignment 1
(Full Score: 50 points)

(Due by 1/31/2011/Monday Midnight at Moodle)

Your name:	Score:
------------	--------

Project I (20 points)

Create a Java program that uses the console to input the first and last name, age, GPA (grade point average), and the date of birth of a student. The program then should output the information to the command-line console in the following format:

<first name><blank><last name><blank><date of birth><blank><age><blank><GPA>

Example: "John Doe 01/01/2010 23 3.65".

Requirements:

1. The data input should be performed using the *java.util.Scanner* class methods.
2. The output should be done by using the *System.out.printf()* method.
3. The format of the output should be:
 - a. 10 positions for the first name.
 - b. 20 positions for the last name.
 - c. 20 positions for the date of birth.
 - d. 4 positions for the age.
 - e. 10 positions with two decimals for the GPA.
 - f. All output fields must be left justified.

Hints:

1. Use Java 6 API specification(via <http://java.sun.com/javase/6/docs/api/> or a link from the course website) for the information on Java 6.
2. Use Java coding guidelines when naming your identifiers and creating class fields. The fields should be *private* or *protected* and appropriate accessor and mutator methods should be provided.

Note:

1. When creating project using a computer in the Computer Science Department laboratory (MH 202/209/210), always make sure that the project and all associated files are on the disk drive that you will be able to access next time. This may be H: drive, USB flash drive, etc. Never store any files on any of the local computer drives as these files will be automatically deleted after you log off.

Turn Over: The 2nd project is on the back

Project II (30 points)

We are going to create a program that will evaluate the value of the mathematical function e^x . The evaluation involves adding terms of series together of a series together till to a certain term (or desired accuracy) is reached.

The value for the function e^x is found by:

$$e^x = 1 + x + (x^2/2!) + (x^3/3!) + (x^4/4!) + \dots + (x^n/n!) + \dots$$

Your program should

1. Ask the user for a value of x that you would like evaluated.
2. Ask the user for a value of the number of terms (the value of the n).
3. (**important!!**) Have a couple of lines of comments in your code (**at the beginning** of the class that contains method `main()`) describing with the use of big-O the time complexity of the algorithms you have used in the implementation of this project, i.e., $O(n)$, $O(n^2)$, ... $O(n \log n)$
 - a. You will receive **ZERO** if you have not completed this step!

Hints:

- a. $5! = 1*2*3*4*5$; $n! = 1*2*3* \dots *(n-1)*n$.
- b. You may use any calculator to test your implementation—you will need to provide a reasonably large n (for example, larger than 8) to test your program.