A) Homework assignment submission requirement/policy

Each homework assignment may contain a set of problems with a due date specified usually one week from the date the problem set is assigned. Assignments will be posted on my website and handed out in class. Each problem in an assignment shall receive a separate grade.

Each homework assignment should be turned in before the specified due time. In case you cannot complete a problem by the beginning of class on the due date, you can take another two days to work on the problem and turn it. The penalty for such a period late submission will be 20 percent. If you hand the problem in two class periods late, the penalty shall be 40 percent for that problem. Beyond one week from the specified due date the problem shall NOT be graded for any reason.

Please assemble all homework in an envelope or folder of your choice. I shall not accept loose homework. The folder should keep the contents from falling out and contain:

1. A clear header including your name (and your partner name in case some lab assignments), assignment number and problem number.
2. A copy of the homework assignment sheet.
3. A printout of all source code and supporting comments.
4. A printout of the output from each program (you may use Alt + PrtSc to copy the console output, and use Ctrl + v to paste it to the project report, say a Word document).
5. A 3 ½ inch DOS-format floppy (or a CD-ROM, or use a USB drive) containing all source code needed to compile and run your program. This diskette must not contain any files unrelated to the problem set. I shall compile and run each program that you submit as well as examine your source code. This source code must be nicely and consistently formatted. Unformatted or difficult to decipher code shall result in a grade reduction for the particular problem being graded.
6. If your problem is one or two class periods late you must clearly state this on top of the first sheet.
7. A self-assessment of each problem. This should indicate whether you believe you have completed the problem successfully. It may also discuss any special difficulties that you have had in solving the problem.

Programs will be graded by compiling and running them on a PC configured like the lab computers. Make sure that the programs can be tested at the DOS console, by the use of javac and java commands. I will NOT use any IDE to grade the programs, though you can use any IDE such as NetBeans to develop the programs. Please verify the contents of your disk before turn in. It is not uncommon to receive disks that contain nothing. That is the grade that is awarded.

If a program does not compile at the DoS console, it shall receive an automatic grade of 0. If a program produces run-time or logical errors you shall receive only partial credit.

* Note: The homework assignment is to be individually completed by yourself. Copying the work of another student whether that work is a homework program or an exam problem is cheating. Obtaining code via the Internet is cheating. You must write your own programs completely and not modify some other student’s work to disguise that the work has not originated from you. It is usually quite easy to see through such disguises. You are always welcome to discuss concepts with fellow students. You must draw a sharp line between discussing a concept and its implementation in a program. The former cooperation is allowed the latter is cheating.
B) Problems using Classes and Objects (total 180pts + 40 bonus pts)

* For full credit your solution must make use of key methods and have minimal code redundancy. Think about your answers.

* You are welcome to drop by my office to discuss your understandings of the exercises, your ideas, or ask for some help. You are welcome to discuss with your classmates about the solutions, but must do coding/debugging by yourself.

* For any problem, try to use “divide and conquer”. If some sub-pieces cannot be accomplished, specify it in the self-assessment and you may receive partial credits for the accomplished pieces.

* For exercises 2-6, try to use this keyword. You should apply the concept of Data Encapsulation. For each class you write, you should write another class to test it. For example, for the exercise 7.1, you will write a Rectangle class, and then write a TestRectangle class (save it to a different file) to test it according to the exercise requirement.

1. Textbook exercise 6.9 (30 pts): Finding the smallest element in an Array

2. Textbook exercise 7.1 (40 pts): The Rectangle class
   (You may refer to Listing 7.5 on page 230-231 of the textbook for reference.)

3. Textbook exercise 7.3 (40pts): The Account class
   (You may refer to Listing 7.8 and Listing 7.9 on pages 240-243 for reference.)

4. Textbook exercise 7.7 (40pts): The Time class

5. Textbook exercise 8.5 (30pts): Occurrences of each digit in a string
   (You may refer to Listing8.2 on page 272 of the textbook for reference. You may use isDigit() method of the wrapper class Character)

6. 40pts bonus: Textbook exercise 7.9, The MyInteger class