Lab Work in MAT 2170

A proper approach to lab work in MAT 2170 is similar to the practicing and training that athletes and musicians undertake. Running laps, lifting weights, shooting baskets, and practicing scales are examples of how athletes and musicians become better at what they do.

Learning how to design computer programs is similar — it takes practice. This is the primary reason why lab assignments are given in this class. Here are some guidelines and expectations for all lab work for this class.

- **Do your own work.** Every exercise solution you turn in for credit should be entirely your own work. You should *never* copy code from another person. A simple rule for this class is that you are not permitted to look at any program that another student is working on, nor should you show anyone else your own work. To act otherwise will be considered plagiarism and is a violation of the EIU policy on academic integrity.

  You may, of course, get help from the tutors and from the instructor. You are also encouraged to talk to others in the class about *general concepts* and *approaches* to solutions, but this should never take the form of Java coding details.

- **Plan your work.** Before arriving in lab, you should have a very *specific plan.* You should never be in the situation that you are entering code "off the top of your head." For most people, this planning involves drawing sketches, designing appropriate algorithms, and writing the actual Java statements needed. It might surprise you to learn that writing the code should probably take no more than 10–15 percent of the total time investment.

- **Check your work.** Choose good test data to gain confidence that your program is doing what you intended. You should also verify your program works correctly if different constant values are chosen or if the window dimensions are changed. After you upload your applets to your EIU web site, make sure they still work they way they should, checking for broken links or incorrect behavior.

- **Polish your work.** Computer programs are often read by other people, so they need to make sense. Make sure you have enough *comments* which explain *what* the program is intended to do, and *how* it accomplishes this. You should:

  – choose meaningful variable and constant names

  – use appropriate indenting and blank lines

  – use correct spelling and grammar

  – include meaningful header and block comments

  – have neatly formatted and informative input prompts and output

  – avoid comments that simply re-iterate program statements; comments should be explanatory and provide an overview of the logic