# Assignment 9

**Topics: Multithreading and review of generics, exception handling and iterators**

**NOTE: exercises with prefix *** are not mandatory**

1. Read about multithreading at: *http://docs.oracle.com/javase/tutorial/essential/concurrency/*

2. Write a program that creates two threads. The first thread's task is to print a certain character c for n times. The second thread's task if to print a certain integer i for m times. To see how thread works run the code several times with large values of n and m.

3. Use the method *yield()* in class *Thread* in the previous program to ensure that after a certain specific number of integer has been printed, a character will be printed. Please notice that (for most of you) this might not work (recall the comments about the method *yield()*).

4. In the previous exercise, use instead the method *join()* to ensure that after a certain specific number of integer values has been printed, the thread that prints out characters will execute until its end.

5. Extend the *Student* class so that when you acquire a new *Student* via keyboard, you have just *N* minutes to finish the input operation. After this time has elapsed print out a warning message.

6. Create a class containing only an integer value as data member (try with both a static integer and a non static integer). Create now one instance of that class and create also 200 threads, each of which adds 1 to that integer variable. Run the code several times. Each time you run it, print out the final value of the integer variable. In the simplest case, you should see that, assuming the initial integer value is set to 0 (or to N), the final value of the integer variable is NOT 200 (or N+200). Why is that so?

7. Fix the logical bug of the previous exercise using thread synchronization