Western Kentucky University
MATH/CS 371: Computational Problem Solving

Spring 2012
Section 001 MWF 08:00am COHH 2117
Section 002 MWF 10:20am COHH 2117
Section 003 MWF 12:40pm COHH 2117
Section 003 MWF 01:50pm COHH 2117

Instructor:	Uta Ziegler	Claus Ernst
Office:	COHH 4108	COHH 4112
Office hours:	W 9:10 – 10:00 am MF 11:30 – 12:15 am T 12:45 – 1:45 pm and by appointment	MF 9:10 – 10:00 am W 11:30 – 12:15 am Th 1:00 - 2:30pm and by appointment
Phone:	745-2911	745-6224
e-mail:	Uta.Ziegler@wku.edu	Claus.Ernst@wku.edu
website:	**Blackboard** http://ecourses.wku.edu	

Description of the course and its objectives: This course aims to improve your skills in problem solving. The problems you will work on involve a wide range of topics that you might encounter in mathematics or any of the sciences. The emphasis in the course will be on computational problem solving, that is on problems that can be tackled with an algorithmic approach and whose solutions involve programming. As a programming environment you will learn to work with *Mathematica*. In the second half of the semester the style of the course will change: lecturing will alternate with guiding you on your project. Teams of two students will work on and complete a project (in exceptional cases a student might be allowed to work on a project by him/herself). Each team is expected to select a different project topic. The project involves problem solving/modeling and a program design with implementation. The majority of programs will be done using Mathematica. However in cases where Java might have advantages, the project can also be implemented with Java.

This course aims at:

- generating excitement for scientific problems which lend themselves to a mathematical/computational approach;
- introducing you to the exciting world of open ended problems, i.e. problems that require deep thought and cannot be worked out in minutes by a strategy that is in a given section of a book;
- developing your ability to investigate challenging problems using mathematical reasoning and computational methods and to communicate your investigation as well as your results; and
- generating joy that comes from the satisfaction of understanding and solving the challenging problems.

Login information for the course website is provided at **ecourses.wku.edu**. (Note that once you log in, all your courses for which there is a blackboard account will show up and our course will be one of those.)

Text: *Mathematica Navigator*, 3rd edition, by H. Ruskeepää.
This text is only a supplement to the course and will serve as a basic reference book for *Mathematica*. We will not follow the book; it is only there to give you the background needed to work with

Mathematica. There is a wealth of examples in the book and you should routinely look and find examples in the book that are helpful to you. The book is quite heavy and we do not want you to bring it to class. Instead install the content of the book on your laptop using the enclosed CD. We expect you to bring the laptop to class every day. In our textbook we will cover some material out of chapters 1-23, with an emphasis on the material in chapters 13 - 18.

Other books: The academy has purchased several books about Mathematica to create a small reference library. The books are available on the bookshelf next to Tim Gott's office. There are only one or two copies for each book and they cannot be checked out for more than an hour or two so that they remain available to all. Consult with the Academy staff about the procedure to check these books out. These books may contain many good ideas that you can use for your project.

What to do during class:
You need to attend every class. Non-attendance will be reported to the Academy. You are expected to participate in discussions of material in class. Your participation will help shape the class and without it this class cannot be successful. We ask pairs of students to work on problems in class together and pick some group(s) to present their work. This helps you practice how to present your ideas. What you present doesn't have to be right – in many problem-solving situations, the first few approaches are wrong – but figuring out that something is wrong and why helps work towards a correct solution. We will use your participation as a small part of your grade. You are expected to take notes in class. And remember, there is no such thing as a "dumb" question. If you are not sure about something, don't hesitate, ASK! Don't be shy about asking questions during class time.

What to do between classes:
You are required to be prepared for each class – that means reading through your notes and working and thinking through what was done in the prior class meeting. Download the files from blackboard if there are some. Make sure you understand the concepts and that you can apply them in similar situations. Read up in your textbook on the *Mathematica* commands used – and check out the examples shown for *Mathematica* commands in the *Mathematica's Documentation Center* or *in our textbook*. If you have many questions or feel somewhat confused or lost come by office hours or set up an appointment. Don't be afraid to admit that there are things that you did not understand. (We expect this to happen to most of you occasionally – it is quite normal and does not mean you are weak.) If you just have a short question, you can ask it at the beginning of class.

Homework: There will be regular homework assignments. To do the homework requires you to think about the questions and problems. This takes time, so start early and work on it every day. We expect that you work on CPS every day. In particular, you need to work on those days when we do not have a class meeting.

The problems you encounter on the assignments will vary greatly in difficulty. You might be asked to solve given problems or to modify algorithms discussed in class to achieve a different purpose or to start investigating a given problem, to share your attempts and observations, and to realize from where the difficulty of the problem arises. For nearly each problem, (most of) you have to figure out what it entails, what tools to use, and how to approach it. This is when you practice problem-solving and it is important to do that for every assignment problem in order to properly learn the material. If a difficulty with a problem arises then ask the instructors for help - asking when you are stuck is an important part of learning! No matter how smart you are, there are problems that are waiting to be cracked by someone smarter than you and your instructors!

Some assignments may require team work. Even though a team turns in work together, each student is responsible for the **entire** work turned in. (This includes making sure that no work was plagiarized.) The purpose of team work is that you can discuss your work with someone else, that you can help each other getting 'unstuck', and to make sure that everybody in the team has thoroughly understood the topics covered.

For the homework, you can use the textbook, other books, the library, the internet, whatever you can find, and talk to your instructors if you are stuck. However there are restrictions. In general, make an attempt to solve a problem on your own before looking for resources on-line, or in the library or where ever. For everyone, problem-solving improves with practice. Remember that very little is gained by reading the solution to a problem before seriously attempting to solve it.

Follow these guidelines/restrictions to avoid getting suspected of cheating:

1. Understand what you read and present the work **in your own words** (if not stated otherwise, all your work has to be turned in 'in your own words.') The 'work' here includes code as well as explanations, solutions, etc. Do not copy code and present it as your own. Do not turn in code without understanding how the code works.
2. Specify the resources you used by providing references (webpages, book titles or article titles, authors and where they are published.) You must provide enough information for the instructors to find the resource without any trouble; this also applies if you use the text book (simply state the page number or the section number e.g. 5.3.3).
3. You may discuss your work with others in the class at a high level (no details, discuss means mostly verbal). (Before turning to someone for help, try to solve a problem by yourself. A lot is learned by you finding approaches to solutions on your own.) However, no written material may be exchanged (except for students working on a team) nor may solutions be worked out together. Every assignment turned in must be developed independently.

Cheating: Cheating on any student work (homework assignments, skills-tests, quizzes, and projects) may result in a failing grade (WKU policy). Turning in a solution that you did not develop without stating the source of the solution is a form of cheating. Plagiarism is a form of cheating. For information on plagiarism and how to avoid it see http://www.indiana.edu/~wts/pamphlets/plagiarism.shtml. Copying and pasting from the Internet (or any other source) without giving due credit is cheating. Turning in a program output that looks correct, but is not generated by your program is cheating. Exchanging solutions to assignments problems is cheating – and so is not safeguarding your work and making it possible for others to turn in your work as their own.

Tests & Quizzes, etc.: There will two skill-tests to be taken in the testing center. There also will be regular quizzes and 'simple tasks'. Make-ups for quizzes will be given only for circumstances beyond your control and the make-up needs to occur before 8:00 am on the day the quiz is returned (which usually is the next class day). It is the student's responsibility to arrange a time to take the make-up quiz. If you miss a quiz with a valid excuse and it cannot be made up any more then it is not included in the average. Simple tasks are tasks to be completed (usually with something to turn in) by the next class meeting and will be recorded as quizzes. Showing on a quiz that you remember/understood what was covered during the last (few) class meeting(s) (including assigned reading) is good. However there may be questions that generalize concepts of the prior class meetings. To earn an A you need to be able to answer these questions as well.

Project: There will be no written final exam. Instead there will a project assigned in the second half of the course. The project will involve much of the methods and tools developed throughout the course. You should look for a partner and a topic for your project. (It will be possible for you to select a partner in a different class – provided it is possible for both of you to attend 371 classes at the same time during the project period!) Each group will work on a different project and topics will be selected through consultation with the instructors. We would like you to come up with a project idea you like to work on. It might be related to research you do or some other interests you have. Implementing a game as project will only be accepted if the team can convince the instructors that it includes significant computational problem-solving aspects (befitting a 300-level course). You need to get serious about the topic of your project by the middle of the semester. Here is an approximate timeline.

Week 7: Settle on good project topic, research feasibility, and turn in written proposal.
Week 8: Start working on your project.
Week 9: Turn in a written project report.
Weeks 10-11: Make a brief presentation about your project to the whole class.
Week 12: Turn in a second written project report.
Weeks 14-15: Final project presentation & final project reports due. Note that week 15 is finals week and the final time assigned to our class will be used as presentation time in addition to the last week of classes.

Note that you will be graded on the progress you make throughout the project period as well on your communication regarding the project (starting with the proposal).

It might happen that the initial topic chosen for you project turns out to be too difficult and needs to be adjusted a bit to make the project workable. This can to be done – but only in consultation and with explicit approval of the modified project by one of the instructors.

Grade: The grade for the course is determined from your grades for the assignments, the quizzes, and the project. In particular, the quizzes count 15% of the grade, the skill-tests count 10% of the grade, homework counts 30% of the grade, the major project is 40% of the grade, and participation is be 5% of the grade. The following grading scheme is used: A: 90-100%, B: 80-89.9%, C: below 70-79.9%, D: 60 – 69.9%, F: below 60%

Late policy: If you turn in an assignment late you will lose 3% of the total score for each day (including weekend days) that you are late. Usually assignments are due at the beginning of class – and an assignment is late if you are late. No credit will be given for late work once it is 7 days late or once the assignment has been returned in class (whatever comes first). No credit will be given for project proposals or progress reports which are more than one day late.

Students with disabilities:
In compliance with university policy, students with disabilities who require accommodations (academic adjustments and/or auxiliary aids or services) for this course must contact the Office for Student Disability, Room 447, Potter Hall. The OFSDS telephone number is (270) 745-5004.

Per university policy, please DO NOT request accommodations directly from the professor or instructor without a letter of accommodation from the Office for Student Disability Services.