

## ENGR 1412 – Course Syllabus – Spring 2015

### INTRODUCTORY ENGINEERING COMPUTER PROGRAMMING

#### **Instructor**

**Dr. Jennifer Glenn**

School of Industrial Engineering and Management

Office: Engineering North 317E

Email: [Jennifer.Glenn@okstate.edu](mailto:Jennifer.Glenn@okstate.edu)

#### **Head Teaching Assistants**

**Suresh Jayaraman (Quizzes, Exams and Projects)**

Graduate Student, School of Chemical Engineering

Email: [sureshj@mail.okstate.edu](mailto:sureshj@mail.okstate.edu)

**Kavin Karappadi (D2L, Communication, Discussion Board)**

Graduate Student, School of Industrial Engineering and Management

Email: [karappa@okstate.edu](mailto:karappa@okstate.edu)

#### **Office Hours**

Dr. Glenn will hold office hours by appointment (only after a confirming email).

You will also be able to get individual help from Teaching Assistants (TAs) and Structured Learning Assistants (SLAs) and you can also post questions on the discussion board.

#### **Textbook**

Computer Programming for Engineers: COURSE NOTES FOR ENGR 1412 - Fall 2012 by Dr. Eric Maase - available from the Student Union Bookstore (This is the version we will use this semester.)

#### **Course Description**

This course is designed to promote constructive thinking, critical analysis and creative problem-solving while learning a computer programming language and programming solutions to engineering problems. Upon completion of the course, you should be able to:

- I. Operate OSU-IT Tools.
  - a. Logon to OSU computer systems and Desire2Learn (D2L).
  - b. Start and navigate Windows Explorer, Excel and Word.
  - c. Edit text and drawings in Word and Excel (copy, paste, etc.)
  - d. Move, copy, attach and print files.

- II. Develop computer-programming skills of utility for student and professional life, such as:
  - a. Interval Halving
  - b. Newton's, Secant, and Bisection methods for root-finding
  - c. Trapezoid rule of integration
  - d. Finite difference equations
  - e. Propagation of uncertainty
  - f. Monte Carlo
  - g. Infinite series
  - h. Vector manipulation
  - i. Polynomial Operations
  - j. Golden Section Optimization
- III. Analyze engineering procedures and identify simple executable stages.
  - a. Do it by hand
  - b. Descartes approach
  - c. Flowcharts
  - d. Algorithms
  - e. Engineering design cycle
- IV. Develop computer programming abilities to effectively:
  - a. Apply math and science to solving engineering problems
  - b. Design solutions and models of engineering processes that meet desired aims / goals
  - c. Understand professional and ethical issues involved in computer simulation and general computer use
  - d. Communicate clearly and professionally with peers
- V. Design, debug, and validate computer programs.
  - a. Understand the user expectations
  - b. Good programming practice
    - i. Header, comments
    - ii. Variable nomenclature, definitions, units
    - iii. Data Echo
    - iv. Program Structure
  - c. Debugging methods
  - d. Error avoidance/trapping
  - e. Validation techniques
    - i. Test against known values
    - ii. Test asymptotic limits
    - iii. Test initialization and cycles
- VI. Use fundamental programming features –VBA environment and language.
  - a. Environment
    - i. Excel and the VBA editor
    - ii. Command buttons
    - iii. Help and Debugging Tools
  - b. Input and Output Operations (commonly called I/O)
    - i. Read/Write to Excel Spreadsheets
    - ii. Input and Message boxes for user output and input
    - iii. Read/Write to external files

- iv. Visual Basic Forms
- c. Programming Language
  - i. Variable data types: Integer, Single, Long, Double, String, Boolean, etc.
  - ii. Assignment statements and operator precedence
  - iii. Built-in functions
  - iv. Conditional (IF-THEN-ELSE) structures and logical operators
  - v. Loop (FOR-NEXT, DO WHILE) structures and program flow
  - vi. Arrays: Vector and Multidimensional
  - vii. Functions and subroutines, transfer of arguments, public/constant/static
  - viii. Object oriented systems / object attributes

## **Course Structure**

Dr. Jennifer Glenn is the instructor for the course. She will be assisted by several teaching assistants (TAs) and structured learning assistants (SLAs). The course will provide you with a variety of learning assistance, including (a) formal lecture sessions, (b) laboratory sections and (c) individual help opportunities, as described below.

### **Lectures:**

You are enrolled in the following lecture section, which meets once/week.

Tuesday      12:30 – 1:20 pm      ES 317

Documentation and materials for lecture topics will be posted to Desire2Learn (D2L) and Top Hat.

### **Lab Sections:**

You are enrolled in **ONE** 2-hour lab section – the day and time vary by section number. (Section 701 is reserved for Honors College students.) Lab sections have several purposes. TAs will provide detailed instruction on topics introduced in the lecture to assist you in grasping basic programming tools. Lab sections also provide in-class time for you to work on programming assignments under the supervision of TAs. Lab sections are scheduled for two classroom-hours and will be that long every week, so expect to be in lab for the scheduled time each week. Attendance will be taken at both the beginning and end of each lab session. During the course of the semester, five quizzes (15-20 minute programming exercises) will be given, each beginning at the start of the lab period.

Two TAs - a primary TA and a secondary TA – will administer each lab section, assisted by a Structured Learning Assistant (SLA). Yes, that's right, a total of three people are in each lab session to help you master the topics in the course! The primary TA presents lab materials and is responsible for grading assignments and preparing and grading quizzes for the section, in accordance with the guidelines developed by the instructor. Questions about lab assignments and lab grades should be directed to the TAs.

**TA Help Sessions:**

Help sessions are blocks of time when TAs are available in computer labs to give you individual help on assignments or other programming questions, and to provide additional time for working on assignments. During help sessions, any ENGR1412 student may use the lab on a first-come, first-served basis. The schedule for each TA will be posted on D2L. You can also obtain assistance from TAs through the online discussion forums in the “ENGR 1412-ENGR COMP PROGRAMMING-SEC TH ALL SECS” course link on D2L.

**Structured Learning Assistant (SLA) Sessions:**

Structured learning assistants (SLA) are undergraduate students who have taken ENGR 1412 previously. They will be present in the lab sections as a resource for you. In addition, SLAs will be present in “SLA sessions” at night between 6:30 and 9:30 pm, which are additional blocks of time for you to receive assistance with coursework. The first and last half hour (at 6:30 and 9:00 pm) of each session will have directed learning opportunities. The rest of the time (7-9 pm) will be for one-on-one help on projects and course content. The schedule for SLA Sessions appears below:

SLA Sessions

Monday	6:30 pm to 9:30 pm	EN 515
Tuesday	6:30 pm to 9:30 pm	EN 515
Thursday	6:30 pm to 9:30 pm	EN 515

**Course Schedule**

The Course Schedule is an up-to-date master table that contains week-by-week information on lecture topics, suggested readings in the course notes, assignment due dates and exam dates. It is posted on Desire2Learn (D2L). While additional topics may be added to the Course Schedule for lectures or labs, the due dates for the Programming Projects, Quizzes and Exams are not expected to change.

**Course Software**

In this course you will learn computer programming practices and principles for engineering using Visual Basic for Applications, VBA. The VBA language is included in the suite of Microsoft Office Programs and the course will teach VBA in concert with Microsoft Excel. You are neither required nor expected to purchase any software; it is available on all university computers, and on most personal computers through a university contract with Microsoft.

There are many computer-programming languages. Those most often used in engineering applications include FORTRAN, C++, and Visual Basic (.NET). You will find that transferring programming skills from one language to another is straightforward, once you learn to work with a computer effectively and efficiently in any one specific language.

**Online Course Management System – Desire2Learn**

Course information, handouts, assignments, and solutions will be posted on the OSU Desire2Learn (D2L) web site. To access it, go to <https://oc.okstate.edu/>, and login with your OSU O-Key username and password. Click on the course link “ENGR 1412-ENGR COMP PROGRAMMING-SEC TH ALL SECS” and from there to specific course items. You are responsible for information posted on the D2L site, and announcements posted are considered to have been seen by all students once the information has been available for 36-hours.

**Computer Access:** The programming language in this course is Visual Basic for Applications (VBA) and is widely accessible in every university or college computer lab, as well as your personal computer, through Microsoft Excel (MS Office 2010/2013). We expect you to work on assignments outside of scheduled labs, and to use TA and SLA availability in the scheduled labs and TA Help/SLA sessions to help you with problems and questions arising from your independent work.

OSU is a “Microsoft campus,” which means that all students have licensed access to MS Office Suite (which includes Excel). To access it, visit [www.okstate.edu](http://www.okstate.edu), and follow the “Software Distribution Center” link on the quick links part of the main page. You need to know your OSU user ID and O-Key password. You may download software only once. You should download file(s) and copy them to a CD for subsequent use.

**CEAT Computers:** Labs and Help Sessions are held in designated computer lab rooms and in some cases classrooms. Computers in the labs are accessible by all students enrolled in engineering courses, using your O-Key login and password. The O-Key login and account setup may be found at <https://okey.okstate.edu>. *Check and confirm that you have computer access **prior** to your first lab!*

### **Method of Learning and Instruction**

In this course, it might be said that the instructor and TAs do not intend to *teach* you anything. Instead, they will introduce you to useful tools and techniques, which you then will learn by practice. Your instructor and TAs will act as guides and coaches, using the assignments and additional examples to help you learn by practicing, by doing programming. The exact path you take to solve any particular problem is going to be your choice.

The TAs are responsible for helping students, but not by providing answers. They will provide direction for self-learning as appropriate to the student. As a result, you should leave the course knowing basic techniques and, more importantly, knowing how to learn new ones.

There will be one 50-minute lecture each week, given in the theory section. Copies of the lecture material, as appropriate, are posted on the course Desire2Learn web site and Top Hat. Lectures will present engineering computer programming topics, often using engineering applications for examples to provide insights into the use of computers in engineering.

The lectures are designed partly to prepare you for a project assignment that is due at least one week later. This allows you plenty of time to work on the projects. You should start working on each new project immediately. If you delay starting a project assignment until the week it is due, then (1) you may not benefit from the lectures, and (2) you are likely to perform poorly on the project assignments.

### **Assessing Student Performance**

Your performance in the course will be assessed on a combination of:

- Four (4) Individual Programming Projects,
- One (1) Group Programming Project,
- Five (5) Quizzes,
- Lab and Lecture Attendance and Participation,
- One (1) Midterm Exam, and
- One (1) Final Exam

for a total of 1,000 points for the course. The Course Schedule contains due dates for each of the above events. The 1,000 points are distributed as shown in the following table, and your grade will be based on the total points earned during the semester:

Midterm Exam	100
Final Exam	200
Quizzes	200
Programming Projects	400
Attendance/Participation	100
Total Points	1,000

### Exams:

- 1) The midterm hour exam (100 points) and a final exam (200 points) are open course book and open notes. You must take these exams as scheduled.
- 2) In exceptional circumstances, such as a documented serious illness or an important school sponsored trip, an absence from the scheduled midterm exam may be excused.
- 3) In order to be excused from midterm exam, you must submit, as early as possible, a written explanation and justification for the absence, together with any documentation (physician's note, Athletic Department request, etc.). This note should also include your name, ID number, date submitted, and your signature.
- 4) Instructor reserves the right to give either (a) a make-up exam or (b) use your percentage score on the final exam to replace the missed exam.
- 5) If notice is not given prior to the exam, or if instructor approval is denied, the missed exam will receive a score of zero.
- 6) Both the midterm and final are comprehensive over all course material covered to that date.
- 7) All students must take the final exam. The final exam will be administered in the lab sections during dead week.

### Quizzes:

Each of the five quizzes is worth 50 points. The quizzes are done on the computer during lab sessions and will be closed book and closed notes; however, you may access application help files (e.g., Excel and VBA). For each quiz, you will write a computer program and/or solve problems on paper, which are graded on how your answers meet specific criteria. Quizzes start promptly at the beginning of your assigned lab period and will last 15-20 minutes. Show up on time! You will know well in advance what programming techniques are covered on each quiz, even though each lab will have a different quiz question. **Your lowest quiz score will be dropped at the end of the semester.** Missed quizzes receive a grade of zero.

### Programming Projects:

There are five Programming Projects. The first two projects are worth 50 points each. The following three projects are worth 100 points each. **Assignments (both hardcopy and softcopy) are due on Tuesdays at 12:00 p.m. (noon).** You are to provide the primary TA of your lab with an electronic copy of your solution (via course drop-box on D2L). You are to submit a hardcopy to the dedicated box in EN 322, the Industrial Engineering and Management main office. You must turn in (a) the

program, (b) a separate write-up, and (c) any requested additional materials, as outlined in the submission instructions for the specific assignment.

The five Programming Projects sum to 400 points, which is a big part of the course grade. You are not allowed to team with other students, except for project #5, which is a group project. The individual programming project solutions must be your sole creative work. It must not appear to be a copy of prior solutions, or solutions of others in the course, regardless of lab section.

There is no provision for making-up a missed programming assignment. Late assignments will not be considered or graded without instructor approval. Missing assignments receive no grade.

### Course Grade:

Your final grade in ENGR1412 will be based on your cumulative point performance, with your letter grade assigned as follows:

Points	Grade
points $\geq$ 900	A
$800 \leq$ points $<$ 900	B
$700 \leq$ points $<$ 800	C
$600 \leq$ points $<$ 700	D
points $<$ 600	F

The instructor reserves the exclusive right to lower the points required for a specific grade.

Historically, no significant grade curve lowering has arisen, so you should not expect one to appear this semester either.

OSU policy requires submission of “6-week” grades in all 1000 and 2000 level courses. The 6-week grade is submitted roughly one-fourth of the way through the semester, and reflects at most only 20-25% of the total graded course work for the entire semester. It is not a half-way indication of performance. In previous semesters, one “6-week” grade of “F” became a final grade of “B”, in another case an “A” became an “F” with both changes being the result of student performance during the remaining 75-80% of the course. The “6-week” grade is an indication of beginning performance only and is not a guarantee of class standing or eventual outcome.

### ATTENDANCE

You are expected to attend every class (lectures and labs) and are responsible for all material, announcements, schedule changes, etc., discussed in class. Both the lectures and labs are important. In addition, University policy requires attendance to be taken in all 1000 and 2000 level courses, and attendance will be taken in lecture and at both the **beginning and end** of each lab period.

#### Attendance/Participation (100 points total)

Regularly attending class is important for you to do well on the quizzes and exams and is our rationale for taking attendance during each lecture and lab session. This will be done through class participation, utilizing the Top Hat Monocle website ([www.tophat.com](http://www.tophat.com)) and app. During lecture sessions, interactive questions will also be administered and graded through this tool.

Make sure you sign up for the *lecture* section. You can access the course directly via the following:

Course Name: **Introduction to Computer Programming**

Direct URL: <http://app.tophat.com/e/723762>

6-digit course code: **723762**

The first week of the semester, your primary TA will show you, during your lab time, how to enroll in Top Hat for your *lab* section. Therefore, each student will be enrolled in two “courses” on Top Hat – the lecture and specific lab section.

Top Hat is a form of classroom technology that you will be able to access using a myriad of technological options at your disposal (your cell phone, smartphone, tablet, laptop, etc.). We will go through a tutorial for this software during the first lab period, but I would highly recommend that you go through the website and familiarize yourself with how it works. Please direct all questions regarding Top Hat to: [support@tophat.com](mailto:support@tophat.com). This includes anything from registration and how to use Top Hat, to checking grades.

The total number of attendance-related points (100) will be distributed as follows:

### **Lecture: 60 points**

There are 13 scheduled lectures this semester (not including the first week of school). You can earn a maximum of six points in each lecture period, and your ten best scores will be used to calculate your total. The six points available in each lecture are earned as follows:

Being present at the lecture: 2 points

Answering two questions: 4 points (2 points/question). The two points/question include 1 point for answering the question, plus 1 point if the answer is correct.

### **Lab: 40 points**

There are 13 scheduled labs for this semester (not including the first week of classes). You will receive 4 points for each lab attended, up to a maximum of 40 points. That means that you could miss three labs - at your discretion – and still earn the 40 points. To receive 4 points, you must be present at both the beginning and end of the lab session.

As you can see from the above, for both lectures and labs you have several days where you could miss class at your discretion (3 lectures and 3 labs) and still earn full attendance points in the course. Since these discretionary days are provided, we will not entertain excuses for additional classes missed, except for those that can be documented: hospitalization, job interview, athletic event, etc. Approval of such cases is at the discretion of the professor and should be submitted to her and she will communicate with the TAs as needed.

The discretionary days are provided to you to accommodate cases where you may have important, but non-critical, reasons for not attending class, they are not intended to encourage you to miss class unless there is a significant reason to do so. Each lecture and lab contains important information, and you should seriously strive to attend every class meeting.

## **GRADE ARBITRATION**

### **Lab Assignments:**

The TAs have authority to change Quiz and Programming Project grades without instructor’s approval, *provided* the change corrects an error in grading and does not conflict with overall lab grading guidelines. If you believe your lab work was graded incorrectly, see your TA not more than one week after the date on which the assignment was returned. After this one-week period, the lab grade is final.

**Grades will be posted on D2L, and no grade will be changed after it has been posted for two**

**weeks. You are responsible for checking the accuracy of your recorded grades.**

### **Exams:**

If you believe a problem on your exam should be re-graded, you must write the reasons why you think so and return this explanation, along with the exam, to your primary lab TA within one week of them being handed back. This explanatory note must include a clear, concise argument of why your grade should be changed. Requests for a grade change based on the argument that the grading was “harsh” or “unfair” are not acceptable since conscientious effort is made to ensure consistency in grading assignments. The instructor will review all exam grade change requests and decide if the grade should change. Exams will be returned to the student in the following lab period.

**Grades will be posted on D2L, and no grade will be changed after it has been posted for two weeks. You are responsible for checking the accuracy of your recorded grades.**

### **Honors Section and Honors Contracts**

Students in the honors section or on honors contract will do all of the work expected of regular students in the course, plus special features intended to broaden student perspectives on computer programming in engineering. While some honors additions to the assignments will require additional technical complexity or additional programming tools, honors topics may also include investigations into human aspects of computers, programs and programming. Some possible topics include identifying pioneers and the significance of their contributions, determining the differences between programming languages, adding a feature to a program, enhancing user interaction or expanding on calculated results.

### **Special Circumstances**

If you have any special circumstances (e.g., health or learning disabilities), please let Dr. Glenn know as soon as possible. Every reasonable effort will be made to make sure that your learning experience is reasonable and straightforward. See the *Syllabus Attachment* (posted on D2L) for additional information. *The earlier you tell us about a need, the more and better solutions we will have at our disposal to satisfy it.*

### **Academic Integrity / Class Honesty**

Academic dishonesty is a particularly serious matter in one’s professional studies and in this course it will be dealt with at the most severe level of punishment. The following is from [academicintegrity.okstate.edu](http://academicintegrity.okstate.edu):

“Oklahoma State University is committed to the maintenance of the highest standards of integrity and ethical conduct of its members. This level of ethical behavior and integrity will be maintained in this course. Participating in a behavior that violates academic integrity (e.g., unauthorized collaboration, plagiarism, multiple submissions, cheating on examinations, fabricating information, helping another person cheat, unauthorized advance access to examinations, altering or destroying the work of others, fraudulently altering academic records, etc.) will result in you being sanctioned. Violations may subject you to disciplinary action including the following: receiving a failing grade on an assignment, examination or course, receiving a notation of a violation of academic integrity on your transcript (F!), and being suspended from the University. You have the right to appeal such charges. Contact the Office of Academic Affairs, 101 Whitehurst, 405-744-5627, [academicintegrity.okstate.edu](http://academicintegrity.okstate.edu).”

Plagiarism is a serious form of academic dishonesty! This includes copying solutions of projects from previous offerings of this class, copying current semester solutions and submitting identical solutions

by two or more students. DO NOT GIVE A COPY OF YOUR PROGRAM OR SPREADSHEET TO ANYONE NOR ACCEPT ONE FROM ANOTHER INDIVIDUAL. Submitted work for all components of the class must be your own (quizzes, exams, projects) and not be identical to previous semesters' work, current semester solutions, or other individuals in the class. See [academicintegrity.okstate.edu](http://academicintegrity.okstate.edu) for more details.

### **Disclaimer**

The instructor reserves the right, when necessary, to alter the grading policy, change examination dates, and modify the syllabus and course content to maximize the overall learning process in the course. Any such modifications will be announced in class. Students are responsible for knowing all changes announced in class or otherwise posted on the D2L web page.

### **P.S. Here are some helpful hints for success in this course**

**Do not study:** This approach to learning, when properly punctuated, becomes “*Do, not study.*” Computer programming and problem-solving in general are arts, crafts. You will not learn how to paint well by memorizing the names of colors, or which bristle type is right for a paint type. You learn to paint by painting, and exploring what you want to create. When you encounter difficulties, then you read the “how to” directions, or ask others for information. Computer programming is very much a creative process by a designer who often “tries this and tries that,” eventually creating tools that work.

**Collaborate:** You are welcome to collaborate, tutor and teach each other, but not copy and share. If you copy or share, you will not really understand the material, you will score poorly on tests and quizzes and, in addition, copying is an academic integrity issue - cheating. Learning, of course, is more important than simply passing a course, as knowledge and skills you learn are useful in future endeavors. You must want to learn.

**Start early:** Assignments often appear easy. Do not be fooled. Each assignment can require several hours of perhaps-aggravating work. As a general rule: ***“10% of a programmer’s time is spent constructing a program (solution) and the rest (90%) of the time is spent making the program work”***.

The best approach is to work a bit, actively, until you hit a snag, and then do something else. While you are otherwise engaged, your mind will continue to search for solutions. Then when you return to complete the computer program, you will have fresh and alternate ideas for solving the problem.

**Six hours is too long:** If you find yourself spending more time than this on any of the first four assignments, you should carefully examine the process(es) you are using to understand the material (lecture concepts, lab assignment) and/or your approach to developing a programming solution (VBA statement usage and syntax). Here are a few hints to help you manage your time.

- Are you effectively using lab and lecture? Attend, take notes, read class notes, and talk/ask questions. Get involved in your classes.
- Have you completely and thoroughly understood the solution methodology? If you have not examined the details (all of the details!) and planned your approach to the problem then you cannot program / teach a computer to solve the problem.
- Juggle one ball at a time. Break up processes / tasks / problems into small, manageable pieces. Make each individual aspect of the program work. Then combine them.
- Learn how to debug a program - the systematic process of discovering and eliminating errors.

*Additional information about university policies and important dates can be found in the Spring 2015 Syllabus Attachment located on D2L under the ENGR 1412 Course Information tab.*