# CS 5090 – Language Design and Implementation

**BGSU Computer Science** **Fall 2015 Syllabus**

**Instructor**: Dr. Robert Dyer
E-mail: **rdyer@bgsu.edu**
Office: HAYES 238
Phone: (419) 372–3469

**Office Hours**: MW 2:45pm – 4:30pm
TR 3:45pm – 5:00pm
OR by appointment (email me)

**Class Meeting Time**: Tuesdays and Thursdays: 6:00pm - 7:15pm, HAYES 117
**Group Work Time**: Tuesdays and Thursdays: 7:15pm - 8:15pm, HAYES 117

**Prerequisites**: CS 2170 (Computer Organization) and CS 3350 (Standard Data Structures and Algorithms), or equivalent.

**Textbook:** We will utilize many (free) online resources in place of a physical textbook. All course texts will be made available on the MIT website 'nb': http://nb.mit.edu/

**Grading**: The final grade will be composed of the following weights.
(The instructor reserves the right to make changes at any time.)

| Assessments | |
|---|---|
| ***Item*** | ***Weight*** |
| Reading Annotations | 25% |
| Problem Set Reflections | 24% |
| Readiness Assurance Assessments (RAAs) | 24% |
| Projects | 27% |

| Grading Scale | |
|---|---|
| ***Range*** | ***Grade*** |
| [90–100]% | A |
| [80–90)% | B |
| [70–80)% | C |
| [60–70)% | D |

**Readings**: Readings are absolutely essential to learning in this course. Every lecture there are readings due, meaning there are 2 readings due each week. Readings will be done on the website 'Perusall' http://www.perusall.com/ which let's you annotate the text being read. **These annotations are required and graded.** The system will automatically grade each annotation. It keeps the 5 highest annotations. Thus, to ensure the best possible score you should aim for 7-10 **good** annotations. Annotations could be you summarizing a piece of text for the class, asking a detailed question, finding problems with the text, etc. Your annotations should be spread out through the whole assigned reading (and not all in 1 small area). You are graded on 3 criteria: quantity of annotations, average quality of annotations, and spacing of the annotations in the text.

**Problem Sets and Reflections**: Problem sets will be assigned for each unit. You are expected to attempt all problems on your own and bring your solutions to the assigned class. I will walk around and grade everyone's solution based on completeness. We will then spend the first portion of the class time in groups, discussing the solutions. Each group will write up a group solution. I will then provide official solutions and allow further discussion. Your solution and the group's solution **are not graded**. It is quite ok to have a wrong solution, as long as you gave the problem an honest attempt!

You are then expected to go home and write a reflection on the problem set due the next class period. This is a chance to outline your own efforts, identify gaps in your own knowledge, and seek additional help. The reflection is graded (using a multiplier based on your completeness score).

**Readiness Assurance Assessments (RAAs)**: To help me gauge where the class is at, we will have in-class assessments at regular intervals. These will utilize Top Hat or be on paper and contain a small number (2-5) of relatively difficult questions. Students will take the assessment in the first half of the class by themselves. You will then collaborate with your group members and re-take the assessment as a group. *The score will be the average of your individual score and the group score.*

**Projects**: One of the goals of this course is for **every student to write a full compiler**. To achieve that goal, we will have several small projects, each building on the effort of the previous project, which will culminate in the creation of a compiler. Projects are done in groups.

**Research Paper**: All 5120 students are expected to write a research paper on a topic of their choice. They will also present the results of the paper to the class. The research component counts as an extra project for grading purposes. Specific details are given later in the course.

**Groups**: This course relies heavily on group work. First day of class we will form groups for the entirety of the semester. You will work with your group on all course projects, the RAAs, and revising the problem sets. **Group members are expected to all contribute equally** and group members will be **rating each other's effort** to help ensure fairness.

**Top Hat**: We will be using the Top Hat (www.tophat.com) classroom response system in class. You will be able to submit answers to in-class questions using Apple or Android smartphones and tablets, laptops, or through text message. You can visit tinyurl.com/TopHatStudentGuide for the Student Quick Start Guide which outlines how you will register for a Top Hat account, as well as providing a brief overview to get you up and running on the system. An email invitation will also be sent to your school email account (if you don't receive this email, you can register by visiting our course website tophat.com/e/224014). Top Hat requires a paid subscription, and is available through the bookstore for a discount.

**Perusall:** The reading annotation website we use is Perusall. To log in, you will go to https://app.perusall.com/ and create a user. You will then provide the course signup code: **ZYEZBTW26E**. For an idea of what type of comments we are looking for, please view this document: http://perusall.com/downloads/scoring-examples.pdf and this document http://perusall.com/downloads/rubric.docx for details on the grading.

**Source Code**: Groups have a choice in programming language, though I recommend either Java or C++. All groups however are required to use Git for version management. Each group will have access to a repository hosted on GitLab https://gitlab.com/ and are expected to use it. The instructor has access to all group repositories and will use it to gauge if all team members are contributing to the project.

**Withdrawal Deadline**: Friday, November 13, 2015. University policy states that after this date, anybody withdrawing from the course will have the grade automatically turn into an F.

**Canvas**: The syllabus, schedule, and course policies will be available on Canvas. Your grades will also be available on Canvas throughout the semester.

**Office Hours and Help**: Please check your Canvas course site, Canvas messages, and your BGSU email **regularly**. *[You may have your Canvas messages forwarded to your BGSU/other email, and have your BGSU email forwarded to another favorite email address, if necessary, but do check it (multiple times) daily.]* I do forward my own Canvas messages to my BGSU email and check my BGSU email multiple times everyday (with rare exceptions). I check BGSU email more often than I access Canvas, so if you need to contact me urgently, use both Canvas and BGSU email, if necessary multiple times. I will do my best to accommodate you

ASAP, even if outside my posted office hours and without appointment. In general, if you need to see me in my office outside of my regular office hours, please make an appointment.

**Attendance**: Students are expected to attend each class and be on time. I do not believe in any specific grade incentives for class attendance, or penalties for absence. More over

**Make-up policy:** If you cannot take an RAA as scheduled, you (or an authorized person, only in case you are unable to do so) must contact me ahead of time with the reason. Note however that any make-up RAA will count 100% toward your score (there will be no averaging with the group RAA score). Make-ups are considered typically for health emergencies only.

**Academic honesty**: All coursework for this class is expected to be YOUR OWN work. The MINIMUM penalty for copying someone's work (including current classmates, students from a previous offering of the course, or postings found on the web) or knowingly allowing someone to copy your work is a zero for the homework/project/exam/paper/presentation. The offense is also reported to the dean of your college. *Turnitin* and *Moss*, plagiarism detection tools, will be used in this course. I will follow the Department's policies and the University's code of academic conduct as defined in the *BGSU Student Handbook*. For details refer to:

1. *Department of Computer Science Academic Honesty Policy* (http://www.bgsu.edu/arts-and-sciences/computer-science/policies-for-current-students.html)
2. *BGSU Code of Academic Conduct* (http://www.bgsu.edu/content/dam/BGSU/student-handbook/documents/Academic-Code-of-Conduct-Chapter.pdf)
3. *The Academic Charter*, section B-I.G (http://www.bgsu.edu/content/dam/BGSU/faculty-senate/documents/academic-charter/B-I-G-Academic-Honesty-Policy.pdf)

**Disability Policy**: In accordance with the University policy, students with disabilities must verify their eligibility through the Office of Disability Services, 38 College Park Office Building, 419–372–8495 (http://www.bgsu.edu/disability-services.html). Contact me as soon as possible this semester to arrange any accommodations needed to assist with your success in this course.

**Religious Holidays:** It is the policy of the University to make every reasonable effort allowing students to observe their religious holidays without academic penalty. In such cases, it is the obligation of the student to provide the instructor with reasonable notice of the dates of religious holidays on which he or she will be absent. Absence from classes or examinations for religious reasons does not relieve the student of responsibility for completing required work missed. Following the necessary notification, the student should consult with the instructor to determine what appropriate alternative opportunity will be provided, allowing the student to fully complete his or her academic responsibilities (*The Academic Charter*, section B–I.F–4.b at: http://www.bgsu.edu/content/dam/BGSU/faculty-senate/documents/academic-charter/B-I-F-Classroom-Related-Responsibilities.pdf).

## Tentative Course Schedule

| Week | Day | Date | Topics / Important Dates | Assigned | Due |
|---|---|---|---|---|---|
| 1 | T | Aug 25 | Introduction; Compilers vs Interpreters | | |
| | R | Aug 27 | Language Specification; EBNF; Regular Expressions | | |
| 2 | T | Sep 1 | Tokenization; Lexical Scanning | **PS#1** | |
| | R | Sep 3 | Top-Down Parsing | **Proj #1** | |
| 3 | T | Sep 8 | Problem Set 1 discussion | | **PS#1** |
| | R | Sep 10 | **Readiness Assessment Activity 1** | | |
| 4 | T | Sep 15 | Scopes and Scoping Rules | **PS#2** | |
| | R | Sep 17 | Symbol Tables | **Proj #2** | **Proj #1** |
| 5 | T | Sep 22 | Problem Set 2 discussion | **Research Paper** | **PS#2** |
| | R | Sep 24 | **Readiness Assessment Activity 2** | | |
| 6 | T | Sep 29 | Type Checking | **PS#3** | |
| | R | Oct 1 | Error Recovery | | **Research Topic** |
| 7 | T | Oct 6 | Problem Set 3 discussion | | **PS#3** |
| | R | Oct 8 | **Readiness Assessment Activity 3** | | |
| 8 | T | Oct 13 | No Class – Fall Break | | |
| | R | Oct 15 | Project 2 Work Day | | **Proj #2** |
| 9 | T | Oct 20 | Intermediate Code Generation | **PS#4** | |
| | R | Oct 22 | Intermediate Code Generation | | |
| 10 | T | Oct 27 | Problem Set 4 discussion | | **PS#4** |
| | R | Oct 29 | **Readiness Assessment Activity 4** | | |
| 11 | T | Nov 3 | Intermediate Code Generation | **Proj #3** | |
| | R | Nov 5 | Procedure Calls | **PS#5** | |
| 12 | T | Nov 10 | Problem Set 5 discussion | | **PS#5** |
| | R | Nov 12 | **Readiness Assessment Activity 5** | | |
| 13 | T | Nov 17 | Program Optimization | **PS#6** | |
| | R | Nov 19 | Optimization Algorithms | | |
| 14 | T | Nov 24 | Problem Set 6 discussion | **Proj #4** | **Proj #3, PS#6** |
| | R | Nov 26 | No Class – Holiday | | |
| 15 | T | Dec 1 | **Readiness Assessment Activity 6** | **PS#7** | **Research Paper** |
| | R | Dec 3 | Project 4 Work Day | | **Research Slides** |
| 16 | T | Dec 8 | Problem Set 7 discussion | | **PS#7** |
| | R | Dec 10 | **Readiness Assessment Activity 7** | | **Proj #4** |
| 17 | T | Dec 15 | **Research Presentations 6:00-8:00pm (Hayes 117)** | | **Presentation** |

**NOTE: If there is a discrepancy between the due dates here and on actual assignments, the one on the assignment applies.**