CS 3060 Fall 2019 – Programming Languages

Examination of a wide variety of programming languages, paradigms, features, and syntaxes through exposure to theory and hands on exercises. Topics covered include static, dynamic, strong and weakly typed, compiled and interpreted, object-oriented, functional and procedural programming, and decision constructs.

Prerequisites Grade of C or better in CS 2020

Contact Information

Instructor	Dr. Robert Dyer
Office Hours	MTWRF 1:30-2:30pm OR by appointment
E-mail	rdyer@bgsu.edu
Office	HAYES 244
Phone	(419) 372–3469

NOTE: The quickest way to reach me is **always via e-mail**. When contacting me via email, if your question is regarding code **always attach the code to the email** and a screenshot of any errors you had.

Textbook There is no required textbook for this course.

Outcomes for the course After successfully completing CS 3060, students will be able to say:

- I can identify differences and similarities across programming languages
- I can describe the differences between different programming paradigms
- I can implement basic algorithms using different programming paradigms
- I can explain the strengths and weaknesses of different type systems
- I can explain the differences between program compilation and interpretation

Grading

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

Assessments

Item	Points Each	Total
Exams *(3)*	100	300
Programming Assignments *(3)*	50	150
Labs *(10)*	10	100
Activities *(10)*	5	50
Total		**600**

Note that there may be more labs and activities than are graded. I will keep only the *highest* in each category, dropping your lowest several grades. However, you must attempt all assessments as any assessment you fail to turn in will *not be dropped*.

Grading Scale

Point Range	Percentage	Grade
540 - 600	90 - 100%	A
480 - 539	80 - 89%	B
420 - 479	70 - 79%	C
360 - 419	60 - 69%	D
0 - 359	below 60%	F

Assessments

Activities

The goal is to give students hands-on experience with the material while I and other students are available to help clarify concepts you may be struggling with. Although these activities are graded, the grade is based on attempting to complete the problem and not necessarily on solving it. Simply give a best effort and submit what you have, and you can earn full points.

For activities, students are encouraged to ask questions and to answer other student questions. Try to avoid giving the solution, as the goal of the activity is for each student to find the solution on their own.

Programming Assignments

There will be several programming assignments. To receive credit for your assignments, they must be submitted on Canvas by the due time. **There are no late submissions allowed.** Partial credit will be given for any completed portion of the assignment, so be sure to submit on time even if you are not finished with the assigment!

Labs

Labs are basically short programming assignments to help you understand the concepts we are learning. To receive credit for your labs, they must be submitted by the due time. **There are no late submissions allowed.** Partial credit will be given for any completed portion of the lab, so be sure to submit on time even if you are not finished with the lab!

For labs, students are encouraged to ask questions and to answer other student questions. Try to avoid giving the solution, as the goal of the activity is for each student to find the solution on their own.

Exams

There is one exam for each programming language studied. Exams consist of a variety of question types, including multiple choice, true/false, short answer, short programming questions, and interpreting code. **Most questions are writing code.**

Technology

Canvas

The syllabus, all assignments, and due dates are posted on Canvas. Your grades will also be available on Canvas throughout the semester. Canvas is the main entry point for this course - everything you need to do is linked and organized from the Canvas course. Always start there!

Plickers

Each student will be assigned their own Plickers card. Plickers cards are 3D barcodes, and depending on the orientation of the card (4 possible sides can face up) you are able to respond to questions with answers A, B, C, or D. This allows quick, interactive feedback from the class. I also use these to quickly record attendance near the start of each class.

Programming Tools

This course requires that you have your own laptop. While I will provide many tips on how to download and install tools for each language, part of the course is learning how to setup your programming environment on your own. So I will give tips and links etc, but won't officially support your personal machines. Note also that the lab computers will not have most of the tools installed.

I highly recommend using Visual Studio Code as your main environment. You can run it off a USB stick, allowing you to use it in the lab. But it won't compile/run the programs on the lab machines. You can download it for free here: https://code.visualstudio.com/download

Course Policies

Withdrawal Deadline

Friday, November 15, 2019. University policy states that after this date, anybody withdrawing from the course will have the grade automatically turn into a F.

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 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. For in person courses, I take attendance at the start of each lecture. For online courses, I check Canvas history to see what each student viewed and how often.

I typically use good attendance as a factor when considering final grades. I reserve the right to penalize students up to 1% of their final grade, per absence, for more than 3 un-excused absences.

Make-up policy

If you cannot take an exam/assessment 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 assessment normally done in groups will count 100% toward your score (there will be no averaging with the team's 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 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 **REMOVAL FROM THE COURSE AND GRADE OF WF**. 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
2. BGSU Code of Academic Conduct
3. The Academic Charter, section B-I.G

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 (https://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).

Classroom Environment, Language, and Behavior Expectations

In order to promote an inclusive and constructive learning environment, demeaning, marginalizing, and otherwise negative language and behavior will not be tolerated in the classroom. Respect and courtesy toward the instructor, classmates, and classroom guests are expected. Language and behaviors that are disruptive, abusive, or harassing may result in disciplinary action as specified by the Student Code of Conduct.

Title IX

Bowling Green State University (BGSU) is committed to providing a safe learning environment for all students that is free of all forms of discrimination and harassment. Sexual misconduct and relationship violence in any form are antithetical to the university's mission and core values, violate university policies, and may also violate federal and state law. Faculty members are considered "Mandatory Reporters" and are required to report incidents of sexual misconduct and relationship violence to the Title IX Coordinator. If you or someone you know has been impacted by sexual harassment, sexual assault, dating or domestic violence, or stalking, please visit www.bgsu.edu/TitleIX to access information about university support and resources.

Tentative Course Schedule

Week	Day	Date	Topics
1	T	Aug 27	Introduction
	R	Aug 29	Haskell introduction
2	T	Sep 3	Haskell - Functions
	R	Sep 5	Haskell Lab - *Hayes 020*
3	T	Sep 10	Haskell - Pattern Matching
	R	Sep 12	Haskell Lab - *Hayes 020*
4	T	Sep 17	Haskell - Data Types
	R	Sep 19	Haskell Lab - *Hayes 020*
5	T	Sep 24	Haskell - map/fold/filter
	R	Sep 26	Haskell Lab - *Hayes 020*
6	T	Oct 1	Haskell - Modules/Type Class
	R	Oct 3	**Exam 1** - *Hayes 020*
7	T	Oct 8	Haskell - I/O
	R	Oct 10	Haskell Lab - *Hayes 020*
8	T	Oct 15	Haskell - Monads
	R	Oct 17	Haskell Lab - *Hayes 020*
9	T	Oct 22	Haskell - Thunks/Trampolines
	R	Oct 24	Haskell Lab - *Hayes 020*
10	T	Oct 29	Scala introduction
	R	Oct 31	Scala Lab - *Hayes 020*
11	T	Nov 5	Scala - Types/Classes
	R	Nov 7	Scala - Generics
12	T	Nov 12	Scala Lab - *Hayes 020*
	R	Nov 14	**Exam 2** - *Hayes 020*
13	T	Nov 19	Scala - Pattern Matching
	R	Nov 21	Scala Lab - *Hayes 020*
14	T	Nov 26	Scala - Traits
	~~R~~	~~Nov 28~~	~~No class - Holiday~~
15	T	Dec 3	Scala - Parallel Collections
	R	Dec 5	Scala Lab - *Hayes 020*
16	M	Dec 9	11:30a-2p **Exam 3** - *Hayes 020*