

COURSE TITLE: Introduction to Computer Programming

YEAR: 2017-2018

INSTRUCTOR: Stacy Dolderer

Phone: 406-268-6366 Email: [stacy\\_dolderer@gfps.k12.mt.us](mailto:stacy_dolderer@gfps.k12.mt.us)

## I. COURSE DESCRIPTION:

This course is intended to introduce computer programming for the student with little or no prior experience and to help students considering a major in computer science to decide whether or not pursue its study. The strategic goals of this course are:

1. Help students gain confidence in their ability to write small programs.
2. Map everyday business problems / tasks into a programming framework.
3. Provide an easier entry into the field than afforded by traditional computer science or engineering programs.
4. Provide students with advantage to compete for jobs by providing competence and confidence as programmers.
5. Allow students from other disciplines to make use of computational methods in their chosen field.

## II. COURSE MATERIALS:

Textbook: Interactive Python: How to Think Like a Computer Scientist

<http://interactivepython.org/runestone/static/thinkcspy/index.html>

Software: Python Integrated Development Environment <https://www.python.org/>

## III. COURSE OBJECTIVES:

Upon completion, the student will:

- Become familiar with programming concepts and methods common to all computer languages;
- Have the ability to transfer these fundamental programming skills to other programming languages;
- Understand how to design simple applications;
- Understand control structures, functions/procedures, arrays, classes, and objects.

## IV. COURSE OUTLINE:

1. Learning a language for expressing computations (Python)
  - expressions
  - variables
  - functions
  - Control Structures
2. Problem definition and solution design
3. Fundamentals of IO

4. Introduction to Data Structures - Arrays
5. Data types – strong vs. weak typing
6. Writing and Debugging a program
7. Learning a set of basic “recipes” - algorithms
8. Scope and visibility
9. Introduction to Object-Oriented Programming – Classes

## V. COURSE CALENDAR:

Week 1: Course Introduction and Python Introduction  
Week 2: Input, Processing, and Output – Assignment 1  
Week 3: Simple Python Data – Assignment 2  
Week 4: Repetition Structures, Turtle Graphics  
Week 5: Turtle Graphics – Assignment 3. Exam 1  
Week 6: Idle Editor, Functions – Assignment 4  
Week 7: Files and Exceptions – Assignment 5  
Week 8: Selection, Decision Structures, and Boolean Logic – Assignment 6  
Week 9: Lists and Tuples - Exam 2  
Week 10: Strings– Assignment 7  
Week 11: Dictionaries and Sets – Assignment 8  
Week 12: Classes and Object-Oriented Programming – Assignment 9  
Week 13: Recursion and Inheritance– Assignment 10  
Week 14: GUI Programming - Exam 3  
Week 15: Final Project  
Week 16: Final Exam

## VI. COURSE EVALUATION:

- Programming tasks      60%
- Quizzes                      15%
- Participation                25%

## VII. MIDTERM GRADES:

*A midterm grade will be posted using the letter grade scale or “S, U, or NA”. Some instructors will use the traditional letter grades as well “A, A-, B+, B, B-, C+, C, C-, D+, D, D- and F or P (Pass)”.*

*S - Satisfactory Progress*

*U - Unsatisfactory Progress*

*NA - Not Applicable*

*X - Not Attending*

*Please talk to the instructor if you have any questions regarding your midterm grade. The midterm grade isn't posted to your official transcript.*

## VIII. Learning Outcomes:

<b>Course Objectives</b>	<b>Aligns with the Following Program/ Degree/Division Outcomes</b>	<b>Type of Course Objective: <i>Introductory, Reinforce, or Emphasize</i></b>	<b>Assessment Tool Used to Determine if Course Objective Has Been Achieved</b>	<b>Great Falls College MSU College Learning Outcomes</b>
Become familiar with programming concepts and methods common to all computer languages;	Understand the fundamentals of computer programming and data structures.	Introductory	Observation Assignments Quizzes Tests	The ability to form strategies to locate, evaluate, and apply information, and know the ethical issues surrounding information and technology.
Have the ability to transfer these fundamental programming skills to other programming languages;	Understand the languages for web and enterprise applications such as Java, Python, PHP, and JavaScript.	Reinforce	Observation Assignments Quizzes Tests	The ability to analyze data, arguments, assumptions, and problems in order to draw conclusions.
Understand how to design simple applications;	Have proficiency in web server administration and application development environments.	Emphasize	Observation Assignments Quizzes Tests	The ability to exercise the skills, competencies and behaviors necessary to succeed in the workplace or at a transfer institution.
Understand control structures, functions/procedures, arrays, classes, and objects.	Understand the fundamentals of computer programming and data structures.	Introductory	Observation Assignments Quizzes Tests	The ability to form strategies to locate, evaluate, and apply information, and know the ethical issues surrounding information and technology.