Course Overview
Introduction to computer programming using Scratch. This course is designed to take you through the essential elements of programming and prepare you to be skilled with Scratch version 3.0. Topics include structure of programming, study of several programming environments used, and projects designed to show knowledge of the content.

Course Outcomes
Below is a list of desired outcomes that students will learn throughout the course.

1. The basics of X-Y axis
   a. Description - In Scratch, the X-Coordinate varies from -240 to 240 and the Y-Coordinate varies from -180 to 180.
   b. Assessment - 1) Your movie should “run” when the green flag is pressed. 2) Move your sprite in any way that you choose around the stage using at least 3 movements that have been discussed and gone over. 3) When the green flag is pressed, your movie should go back to the beginning.

2. Basic movement, making your sprite move forward and spin
   a. Description - Using basic operations, make your sprite move from one side of the stage to the other, turn around and move back across the stage.
   b. Assessment - 1) Your movie should “run” when the green flag is pressed. 2) Move your sprite in any way that you choose around the stage using at least 3 movements that have been discussed and gone over. 3) When the green flag is pressed, your movie should go back to the beginning.

3. Changing backgrounds
   a. Description - You will add, create, and change your background to match the scene you are creating.
   b. Assessment - 1) Your movie should “run” when the green flag is pressed. 2) Create an animation where your sprite changes costumes and backgrounds to alter the way that sprite is perceived. 3) Add sound or music that goes well with your movie. 4) When the green flag is pressed, your movie should go back to the beginning.

4. The role of costumes and changing costumes in animation
   a. Description - You will be able to edit your sprite’s design (the way it looks) in order for your movie to look unique and make sense.
b. **Assessment** - 1) Your movie should “run” when the green flag is pressed. 2) Create an animation where your sprite changes costumes and backgrounds to alter the way that sprite is perceived. 3) Add sound or music that goes well with your movie. 4) When the green flag is pressed, your movie should go back to the beginning.

5. **Adding sounds and making music**
   
a. **Description** - Add music, sounds, or speaking to your movie, to make it unique and interesting.

   b. **Assessment** - 1) Your movie should “run” when the green flag is pressed. 2) Create an animation where your sprite changes costumes and backgrounds to alter the way that sprite is perceived. 3) Add sound or music that goes well with your movie. 4) When the green flag is pressed, your movie should go back to the beginning.

6. **Loops and motions**
   
a. **Description** - You will be able to build a proper loop to keep a sprite or event happening for an extended period of time.

   b. **Assessment** - 1) Your movie should “run” when the green flag is pressed. 2) Create a design, by having your sprite do the same moves over and over again. 3) When the green flag is pressed, your movie should go back to the beginning.

7. **Coordinates & sensing**
   
a. **Description** - Be able to send a sprite to desired coordinates during an allotted time. Also, be able to make your sprite sense if it is touching any specific items, colors, or other sprites to make a certain event happen.

   b. **Assessment** - 1) Your movie should “run” when the green flag is pressed. 2) Create a maze game to get from one point on the stage to the other. 3) If you touch (Sense) a certain color that you shouldn’t, you start over. 4) When the green flag is pressed, your movie should go back to the beginning.

8. **Drawing with sprites using the pen tool**
   
a. **Description** - You will be able to put a pen down, draw, put a pen up, move, etc., in order to create a pattern or design.

   b. **Assessment** - 1) Your movie should “run” when the green flag is pressed. 2) Your sprite should lower the pen so the end result is something that you created using repeating elements. 3) When the green flag is pressed, your movie should go back to the beginning.

9. **Creating animated stories**
   
a. **Description** - Be able to tell a story, by creating and manipulating sprites, sounds, and music to follow a text.

   b. **Assessment** -
10. Creating and using variables
   a. **Description** - Using blocks and creating uniquely named variables, you will create placeholders for data being used in your movie.
   b. **Assessment** - 1) Your movie should “run” when the green flag is pressed. 
      2) You should create a game where your main sprite interacts with other sprites to accomplish a final goal. 3) Use at least 2 variables to keep track of things (score, lives left, level, etc..) 4) Your user should know if they have won, lost, or what their final outcome is. 5) When the green flag is pressed, your movie should go back to the beginning.

11. Creating games
   a. **Description** - Be able to create a game that has a beginning, a goal, and an ending outcome that ends the game all while challenging users.
   b. **Assessment** - 1) Your movie should “run” when the green flag is pressed. 
      2) You should create a game where your main sprite interacts with other sprites to accomplish a final goal. 3) Use at least 2 variables to keep track of things (score, lives left, level, etc..) 4) Your user should know if they have won, lost, or what their final outcome is. 5) When the green flag is pressed, your movie should go back to the beginning.

**Intermediate Outcomes**

Below is a list of intermediate outcomes that may be used throughout the course.

1. **Controlling and motioning blocks together.**
   a. **Description** - Be able to combine multiple blocks together (Motion, Look, Sound, etc..) to achieve your desired outcome for that block

2. **Using the broadcasting block.**
   a. **Description** - Make two sprites converse with each other and keep the timing accurate for readers.

3. **Using the if/else block.**
   a. **Description** - Create an outcome that your sprite will do if one thing happens and have them perform another if a different outcome happens.

4. **Knowing the difference between logical operators.**
   a. **Description** - Know what the And/Or/Not operators are, what they do, and when to correctly implement them into your movie.

5. **Implementing logical operators and if/else together.**
   a. **Description** - Create a complex set of movements combining your if/else statement with a logical operator.

6. **Using variables + variables+motions**
   a. **Description** - Use multiple variables, sequences, and motion sets to make your sprites create an exciting and unique experience for your audience.
Course Structure/Learning Activities

Weekly activities for this course will typically consist of the following items:

Class Lecture/Programming videos
- Hosted by the Teacher or posted online.
- Lecture/Video reflections should be completed and submitted by the upcoming due date

Paired-programming activity
- This activity asks you to apply the materials from the lecture or the videos from the week.
- You will typically be asked to work with a partner.
- We will discuss software that will allow you to work online if you are unable to meet outside of class
- In my sections of this course this is typically completed during class time on Wednesday.
- It is suggested that these are submitted to the assignment link no later than Sunday night.

Individual programming activity
- Now that you have a little more practice, this activity will ask you to work on your own to write some code.
- There will be ways for you to submit your Scratch code for peer review.
- If your code does not pass review/testing you will be allowed to continue to work on it until it does.
- These will typically be completed and submitted by the due date (Usually Friday)
- It is suggested that corrections or additions are completed by Sunday.

Peer Reviews/Code Walkthroughs
- Each week it is expected that you have completed your individual programming assignments and they are submitted by Monday at the start of class.
- After that, we can conduct additional peer-reviews in Scratch or schedule individual meet ups after class during a time that works to meet.
- Every Monday in class will consist of code walkthroughs from the previous week. These will be run by the teacher and students.

Additional items/Unit Check for understanding
- Most activities in this course can be attempted multiple times.
• When completing programming activities, programming assignments, peer reviews, and class discussions, you may resubmit coding assignments multiple times up to the unit deadline.
• There will be a check for understanding at the end of each unit. If you receive a grade that you feel is unacceptable and you want to retake it, you may meet with the teacher to review/relearn the material and then take a revision of the unit check for understanding.
• The grade you earn on the original Unit check for understanding will be final unless a retake earns you a better grade. If for some reason you do worse on the retake, both grades will be averaged together.
• The final exam is the only activity that may NOT be attempted more than once.

Course Grading
You will earn many grades/scores throughout this course. Each of these are assigned points and will be based on the idea that an A is 95% and above, and so on (See Score Cutoff Grade Scale for reference)

1. Unit Activity Grades
   - Each unit is graded independently based on activity specifications.
   - There will be many unit activities and plenty of opportunities for students to be able to showcase their talents and understanding of the material.

2. Competency Demos
   - These are sort of like "midterm" exams.
   - They will be given to students to assess their understanding of the current units topics.

3. Final Exam
   - You must take this exam in the presence of a teacher.
   - Failure to take the final exam results in an F for the course.

At the end of the semester you will have had the opportunity to earn numerous points. Your overall grade will be based on the sum of these scores and the following hard cutoffs.

Score Cutoff Grade Scale:

A   100%-95%
A-  94%-90%
B+  89%-87%
B   86%-83%
B-  82%-80%
C+  79%-77%
Conclusions/Discussion
This course will teach you how to program in Scratch, an easy to use visual programming language. More importantly, it will introduce you to the fundamental principles of computing and it will help you think like a software engineer. Introduction to Programming is designed to give you the foundational skills that will prove important for any type of programming you want to do. This course, which includes lectures, videos, lab exercises, and programming code exercises will prepare you for entry-level jobs in development—or simply make you comfortable with code and confident if you choose to take an advanced study course in programming.