Intro to Programming Class  
Donna Huston and Katy Fleming

Course Overview
This class will give a student the basic overview of computer programming using both Scratch and Python as the main methods of teaching and learning. We got our standards and abilities from the Iowa Computer Science Standards.

Course Outcomes

Outcome 1  
Block coding in Scratch  
Students should be able to…

3B-AP-10 Use and adapt classic algorithms to solve computational problems.
3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-12 Compare and contrast fundamental data structures and their uses.
3B-AP-13 Illustrate the flow of execution of a recursive algorithm.
3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

Outcome 2  
Programming with Python  
Students should be able to…

3B-AP-10 Use and adapt classic algorithms to solve computational problems.
3B-AP-11 Evaluate algorithms in terms of their efficiency, correctness, and clarity.
3B-AP-12 Compare and contrast fundamental data structures and their uses.
3B-AP-13 Illustrate the flow of execution of a recursive algorithm.
3B-AP-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.
3B-AP-15 Analyze a large-scale computational problem and identify generalizable patterns that can be applied to a solution.
3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.

Intermediate Outcomes: Scratch

Intermediate Outcome 1: Simple Animation  
Assessment:  
The student will complete an animation in class using the following guidelines:
1. Run when the green flag is pressed
2. Move the character around the screen in a sequence of moves.
3. Consist of a different series of animations, one right after the other.
4. Pass the "places everyone" test. That is no matter when the green flag is pressed your program should run correctly.

Learning Activities:
1. Demonstrate how scratch works on the projector/screen and go over the different blocks and how they work
2. If necessary, use multiple days to go over the different categories of blocks in scratch and what they do
3. Programming problem - how to get the sprite from one section of the screen to another
4. Programming problem - change things within Scratch; size, costumes, backgrounds, etc.
5. Programming problem - using x, y coordinates to get sprites to certain places in Scratch
6. Programming problem - using repetitive motion
7. Students will be providing feedback by commenting on each other’s work.

**Intermediate Outcome 2: Storytelling (include 2 sprites, sound, and words)**
Assessment: Students will create an animation of a story that they choose (can be a real-life story, nursery rhyme, etc.) using the tasks that they learned in scratch. The animation that they create needs to have the following:

1. Sequence several "actions" to "tell" a story/nursery rhyme when the green flag is pressed
2. Use at least three different sprites.
3. Use at least one sprite that comes from "outside" of Scratch (a personal image or something you found on the Internet).
4. Have at least one sprite send a "message" to another sprite which triggers an action.
5. Include at least five lines of "dialog"
6. Pass the "places everyone" test. As explained in the previous assignment, this means that no matter when the green flag is pressed - even in the middle of telling running/telling it - the story will run properly.

Learning Activities:

1. Introduce broadcast/receive in class in a code walkthrough
2. Programming problem - make a knock/knock joke
3. Show students how to bring in sprites/how to change sprites
4. Show students how to add sound effects
5. Show students how to control speech bubbles
6. Programming problem - how to update another person’s story

**Intermediate Outcome 3: Interactive Stories**
Assessment: Students will create a story with questions that a user answers and the code gives an outcome based on the input.

1. Your program should "run" when the green flag is pressed.
2. When your program runs it should conduct a proper "places everyone."
3. Ask a human user at least four different questions. Introduce yourself to the user while incorporating the answers
4. You should include at least one each of: Open ended question (name, favorite color, etc), Choose between two options (yes/no, boy/girl, cat/dog), Choose from three or more options
5. You should use conditionals to respond differently (and appropriately) to each of the questions that offer option choices.
6. Use several sprites/backgrounds/costumes that go along with your introduction of yourself (even better if they adapt based on the user's answers).

Learning Activities:

1. Demonstrate how to use the ask blocks and how to set variables
2. Demonstrate how to use conditionals
3. Programming problem - edit a previous nursery rhyme to have user input
4. Programming problem - make a mad libs

**Intermediate Outcome 4: Making Games**
Assessment: Students should be able to make a video game that calculates different variables and keeps track of several different kinds of variables simultaneously.

1. Use at least three different sprites in your program
2. One of these should be a human controlled player which uses either mouse or keyboard actions to control the main character.
3. The other two (or more) should be Non-Player Characters which are computer controlled players that can be either good or bad.
4. Set up your game so that the main character "interacts" with the other sprites.
5. At least one of your characters should have multiple costumes that are used in some capacity.
6. Use at least two variables that keep track of two unique things.
7. There should be differences in the way the game plays at different times based on those variables.
8. There should be a clear sense of "winning" or "losing" both in how the game is played AND how the game reacts to what you do when you play.

Learning activities:
1. Demonstrate how to make a character in the game move using the arrow keys
2. Demonstrate how to make another sprite move independently
3. Demonstrate nested if- statements
4. Demonstrate how to change a variable based on if-statements
5. Demonstrate how to make a sprite repeat action after changing variable
6. Demonstrate how to changing variable results in complexity of the game
7. Programming problem - how to make a pick-up game
8. Programming problem - plan a game
9. Demonstrate how to make a sprite follow the mouse
10. Demonstrate problems that come with multiple sprites doing multiple actions
11. Demonstrate how you need to have a game ending
12. Programing problem - making a feeding frenzy game

**Intermediate Outcome 5**: Create Your Own Functions
Assessment: Create a picture in scratch using the pen and function blocks. Use a picture that has many repeating shapes inside the image. Also use shapes that use different parameters within the image.
1. "Run" when the green flag is pressed.
2. Your program should always work correctly no matter when the green flag is pressed.
3. You should move the sprite and raise and lower the pen so that the end result is some scene that you created containing some repeating elements.
4. Your program should make use of at least three blocks (functions) that you defined in the "Make Blocks" menu.
5. At least two of these blocks should be new for this program and not simply copies of ones we made in the lesson materials.
6. At least two of these blocks should include parameters which are used mathematically in the function.
7. All three of these blocks should be used at least twice in the creation of the scene

Learning activities:
1. Show students how to get function blocks/how to alter them
2. Show students the pen feature and how it works
3. Programming problem - make a square
4. Show students how to change parameters
5. Programming problem - how to make smaller and larger squares
6. Create a function block for a scene that will be called in your program

**Intermediate Outcome 6**: Trivia Quiz
Assessment: Students will create a trivia quiz based on factual information that they compile themselves. The quiz will need to have a solid endpoint. The quiz will need to have different intactions if the answer was right or wrong. There will also need to be a measure of score.
1. Run when green flag pressed
2. Setup is correct
3. Multiple choice questions of 5 or more
4. Use lists
5. Wait for user input
6. Provide feedback

Learning activities:
1. Demonstrate how to bring in a list/make a list
2. Demonstrate random question asking
3. Demonstrate how to relate question list to answer list
4. Demonstrate how to make a response for a good response
5. Demonstrate how to make a response for a bad response
6. Programming problem - make a state capital quiz

Intermediate Outcomes: Python

Intermediate Outcome 1: Understanding Python
Assessment: Students will demonstrate understanding of learning activities by the following:
1. Students demonstrate understanding by doing a simple print statement.
2. Create a mad lib for user input generation
3. Debugging assignment
4. Math problem

Feedback: Code walkthrough, pair coding for debugging

Learning Activities:
1. Print statements
2. Output
3. Input
4. Syntax
5. Variables
6. Strings and integers and floats

Intermediate Outcome 2: Formulas
Assessment: Students will be able to show knowledge about formulas by calculating complex math formulas.
Develop a math problem about toilet paper usage and amount of people, types of people, ages, changing of the roll, per bathroom, etc.

Learning Objectives:
1. Basic calculations
2. Strings and integers and floats
3. Operators
4. Order of operations
5. Modular division
6. Built-in functions (max, min, random, etc.)

Intermediate Outcome 3: Functions
Assessment: Room Area-
For this lab you will find the area of an irregularly shaped room with the shape as shown above.

Ask the user to enter the values for sides A, B, C, D, and E and print out the total room area.

Remember the formula for finding the area of a rectangle is length * width and the area of a right triangle is 0.5 * the base * height.

Please note the final area should be in decimal format.

Feedback: The lab gives error statements for debugging

Learning Objectives:
1. Define keyword
2. In parenthesis define parameters
3. Use correct syntax
4. Code function formulas, etc.
5. Return function
6. Call function

Intermediate Outcome 4: Boolean Statements
Assessment: Students will be able to develop code to find multiple results based on information entered by user input.
Example: Leap Year and BMI Formulas

Feedback: Debug and code Walkthrough

Learning Objectives:
1. True or False
2. simple if’s
3. and, or, not
4. if-else
5. else-if
6. algorithms
Intermediate Outcome 5: Loops

Assessment:
Have the students write a game called “Play Pig”

Each turn, a player repeatedly rolls a die until either a 1 is rolled or the player decides to "hold":

- If the player rolls a 1, they score nothing and it becomes the next player's turn.
- If the player rolls any other number, it is added to their turn total and the player's turn continues.
- If a player chooses to "hold", their turn total is added to their score, and it becomes the next player's turn.

The first player to score 100 or more points wins.

For example, the first player, Donald, begins a turn with a roll of 5. Donald could hold and score 5 points, but chooses to roll again. Donald rolls a 2, and could hold with a turn total of 7 points, but chooses to roll again. Donald rolls a 1, and must end his turn without scoring. The next player, Alexis, rolls the sequence 4-5-3-5-5, after which she chooses to hold, and adds her turn total of 22 points to her score.

Feedback: Lab gives error messages for debugging. Pair share and run each other's programs.

Learning Objectives:
1. while loops
2. count variables
3. ways to end a loop
4. guessing game