

Introduction to Computer Science

Course Syllabus



Developed by:
Pat Murakami
patricia.murakami@wright.edu
Summer 2016

TABLE OF CONTENTS

Table of Contents.....	2
1. Introduction to Computer Science.....	3
1.1. Introduction	3
1.2. Materials	4
2. SYLLABUS	4
2.1. Resources	5
2.2.1 QUARTER 1.....	5

1. INTRODUCTION TO COMPUTER SCIENCE

1.1. INTRODUCTION

The intent of this syllabus is to provide a practical guide that educators can follow for sixth grade that is a quarter-long introduction to computer science class. This course provides exposure to block coding using Tynker® and Scratch®. Students will follow a design phase inclusive of flowcharts, block code, and peer feedback. The culminating project involves students programming Ozobots® for specific functions. This Project Based Learning (PBL) project will provide students with the opportunity to work through the design phase.

Differentiation will be addressed through pre-assessment and student need while working through the various software programs. Voice and choice will give students the option to choose games within Tynker® and Scratch® based on ability and interest. While working through Scratch®, higher-level students will program using a language in place of block coding. This is scaffolded so students begin with block code, progress to viewing programming language while using block code, and then write code.

Formative assessment monitors student learning using student samples of programs, reflections, flowcharts or pseudocode, observations, weekly check-ins, peer critique, and logbook entries. Summative assessments include quizzes and the culminating project. Quizzes cover terms and physical use of the software to show mastery.

Online logbooks provide a location for students to record information obtained during the course. The logbooks will provide a structure to record struggles, strengths, questions and general thoughts. Logbook entries will be assessed on a weekly or biweekly basis. A rubric will assess each section of the logbook entries for completeness and accuracy of information.

In order to allow students to progress at their own pace the classroom will need to have a visible question board. This tool provides a central location for the students to ask questions, get answers, and obtain assignments and rubrics. This way the students can become more independent learners by checking the board, then a peer, and lastly the teacher. A collaborative culture is necessary for students to work and this process of using a class question board encourages collaboration. The board will also provide students with assignments to work at their own pace once they become acclimated to coding.

Students will have the opportunity to continue class discussions or ask classmates questions using an online discussion board. This can be accomplished using classroom.google.com. The students can post questions or comments for peers to respond. This will provide an additional format for seeking out assistance from peers.

Students will practice a Growth Mindset using critique. The design process includes revision and feedback so students can learn from mistakes. Revisions and feedback will take place throughout the process and when students finish coding a game. It is important to designate time during the design process to give and receive feedback to ensure the work is strong.

1.2. MATERIALS

- iPad for Ozobots® and Tynker® games
- Lenovo Netbooks for Tynker® and Scratch® games

2. SYLLABUS

The course teaches the essential ideas of Computer Science looking at patterns of code. Students will begin with manipulating icons and progress to visual coding language. Course concepts include problem solving techniques, program design, functions, loops, and conditionals. The course uses computer programming challenges from Tynker®, Scratch®, and Ozobots®. The purpose of the course is to introduce computer science practices and concepts.

This course is divided into three sections to introduce multiple coding systems.

Section 1:

- Tynker® will introduce block coding through gaming

Section 2:

- Scratch® will help students create the block coding for Sprites and objects using sound, movement, color, etc.

Section 3:

- Combines block coding used in the previous two sections with Ozobot® Bits that move from the written code. Writing code for an object to move light up, and wait gives the students an interactive experience.

The course ends with a PBL project using the design process.

Goals of this course:

1. Help students feel confident in their ability to write small block programs.
2. Prepare 6th grade students for entry into computer science pathways in 7th and 8th grades.
3. Learning about the process of writing and debugging a program

Upon completion of this course students will be able to:

1. Explain the process of developing a program
2. Think critically about how to solve a problem
3. Recognize debugging as an application of the design process
4. Create a PBL project using Ozobots®

2.1. RESOURCES

The resources used during this course are downloadable from the internet. Students use the assigned resources which also provide the opportunity to extend learning using additional programs. Tutorials to introduce each section are provided below:

- 1) Introduction to Tynker®: <http://www.tynker.com/myclasses/lesson-preview?classid=575ec3a465e4f25d248b456e&chapterid=546fb14684aafac1200004c>
- 2) Introduction to Scratch®: <https://scratch.mit.edu/help/videos/>
- 3) Introduction to Ozobot®: <http://www.Ozoblocky.com>

2.2.1 QUARTER 1

The flow of the daily activities is flexible based on student progress and needs.

Day 1 **Class Activity** – Dash and Dot introduction

Instruction – Course expectations and syllabus. Dash and Dot are robots that work using block coding. These robots are used as a hook for learning about computer science. You do not need to purchase these robots. A free online game could be used as a hook as well. To learn more about Dash and Dot go to this website:

<https://www.makewonder.com/dash>

Assessment – Written coding pre-assessment

Day 2 **Class Activity** – Introduce online logbooks (set up file folder and document)

Instruction – Introduce students to correct procedures for using iPad. Students will investigate how Dot and Dash work and manipulate robots.

Logbook Entry – Notes on iPad use and observations and procedures for Dot and Dash

Day 3 **Class Activity** – Discussion of careers in computer science

Instruction – PPT (Stop at #24)

Resource(s) – Computer Science PPT:

https://code.org/files/computer_science_advocacy.pptx

Logbook Entry – Notes on computer science careers – education & job opportunities

Day 4 **Class Activity** – Introduce block coding activity. Begin using Tynker®. Students will work at own pace through games.

Instruction – Students will view introductory video on Tynker®. Give students paper cut outs of block coding and have them work create a program for walking to door and shutting it. Review the work and share through a Gallery walk. Then students will play Candy Crush® game.

Resource(s) – Tynker® introductory video: <http://www.tynker.com/myclasses/lesson-preview?classid=575ec3a465e4f25d248b456e&chapterid=546fb14684aafacf1200004c>
Candy Quest: <http://www.tynker.com/hour-of-code/2014?s=candy-quest>, Code Monster: <https://www.tynker.com/courses/codemonsters/mapCodemonsters>

Logbook Entry – Notes & struggles and concerns

Day 5 **Class Activity** – Students will continue to work through the Tynker® games.

Instruction – Share out what students learned about the programs (allow students to use information from notes). Discuss movement, etc. and how the students think this movement occurs. Put Candy Quest on white board and go through some of the games to discuss movement. Address concerns and questions and then students will continue to work through the remaining games.

Resource(s) – Dragon Dash: <https://www.tynker.com/hour-of-code/2014?s=dragon-dash>, Tynker® & Minecraft: https://www.tynker.com/minecraft/Tynker_Minecraft

Logbook Entry – Notes & struggles and concerns. Students will also record patterns they see in the different games.

Day 6 **Class Activity** – Students will continue to work through the Tynker® games.

Instruction – Share out what students learned about the programs (allow students to use information from notes). Discuss any patterns observed while playing the games. Students will continue to work through the remaining games.

Resource(s) – Code Commander: <https://www.tynker.com/courses/>, Debugger: <https://www.tynker.com/hour-of-code/2014?s=debugger>

Logbook Entry – Notes & struggles and concerns.

Day 7 **Class Activity** - Introduce Design Process used during this course. Steps: Clarify problem, define the output (expected results), develop logic (flowcharts/pseudocode), write code, and test/debugging.

Instruction – Introduce steps of design process using Scratch® software. Model the steps using “Animate Your Name” tutorial. Then have students work through the 11 tutorials to view and practice.

- 1) Getting Started with Scratch®
- 2) Animate Your Name
- 3) Make It Fly
- 4) Make Music
- 5) Race to the Finish
- 6) Dance, Dance, Dance
- 7) Create a Pong Game
- 8) Catch Game
- 9) Hide-and-Seek Game
- 10) Create a Virtual Pet
- 11) Favorite Things

Resource(s) – Scratch project at:

https://scratch.mit.edu/projects/editor/?tip_bar=home and Scratch® Cheat sheet <http://www.dummies.com/how-to/content/scratch-for-kids-for-dummies-cheat-sheet.html> provide a print out of the cheat sheet.

Logbook Entry – Notes from tutorials, struggles and strengths while working through tutorials.

Day 8 **Class Activity** – **Students will work through each activity for 6 days.** Debrief on previous days experiences. Review any of the steps from Scratch® that students need clarification on and discuss similarities and differences between Scratch® and Tynker®.

Instruction – Students will continue to work through the components of Scratch®. The instructor will circulate around the room to answer questions and prompt understanding.

Resource(s) – Scratch project: https://scratch.mit.edu/projects/editor/?tip_bar=home

Logbook Entry – Notes from tutorials, struggles and strengths while working through tutorials.

Day 15 **Class Activity** - Debrief on previous days experiences. Review any of the steps from Scratch® that students need clarification on and discuss similarities and differences between Scratch® and Tynker®. Create a VENN Diagram on the board to incorporate discussion.

Instruction – Students will continue to work through the components of Scratch®. The instructor will circulate around the room to answer questions and prompt understanding.

Resource(s) – Scratch project: https://scratch.mit.edu/projects/editor/?tip_bar=home

Logbook Entry – Notes from tutorials, struggles and strengths while working through tutorials.

Day 16 **Class Activity** – Debrief on tutorials and discuss any concerns or questions gathered in logbook. Discuss design process beginning with the problem to solve/code.

Instruction – Model using “I do, We do, You do” the design process. Begin with the problem. Talk out loud as if students are hearing your thoughts in your head. Then introduce flowcharts/pseudocodes with a paper activity. Model how to make a peanut butter and jelly sandwich using a flowchart. Use stencils for the shapes. Discuss any concerns and/or questions and then have students create their own flowchart of brushing their teeth. Each student will work individually and then share their steps with a partner to test and debug. Explain the debugging process. Wrap up with information learned from flowcharts/pseudocodes.

Logbook Entry – Write down flowcharts/pseudocodes with an explanation of what the shapes represent.

Day 17 **Class Activity** – Debrief from previous lesson and address questions and concerns. Practice writing code in logbook to test and debug.

Instruction – Write block code in logbook from one of the activities from previous lesson. Once the code is written, share with a partner to test and then debug.

Logbook Entry – Record code and reflect on struggles and strengths.

Day 18 **Class Activity** – Introduce Scratch® project and rubric.

Instruction – Students are to write a program that incorporates commands they have learned while programming in Scratch®. The students will follow the design process:

- Problem: Create a problem that exhibits knowledge learned from Scratch® games.
- Criteria: Choose a character, backdrop; choose appropriate events, motions (at least 2), sound, and one other element for the program. Develop logic using either flowchart or pseudocode. Write code using Scratch®.
- Test and debug the program.

Resource(s) –Scratch project: https://scratch.mit.edu/projects/editor/?tip_bar=home and Scratch® cheat sheet

Logbook Entry – Record criteria, constraints, and questions.

Day 19 **Class Activity** – Students will work for 2-3 days to create, test, and debug game using criteria stated in rubric.

Instruction – Students will use Scratch® cheat sheet and website to create project.

Resource(s) – Scratch project: https://scratch.mit.edu/projects/editor/?tip_bar=home and Scratch® cheat sheet

Logbook Entry – Record notes and ideas along with flowchart and/or pseudocode.

Day 22 **Class Activity** – Students will share games through a Gallery Walk. Peers will evaluate the game for playability, interest, and meeting criteria.

Instruction – Gallery Walk and record notes from peer’s game.

Logbook Entry – Record ideas gathered from peers, concerns, and questions.

Day 23 **Class Activity** – Students will reflect on Scratch® project.

Instruction – Students will reflect in their logbooks on the process of creating the project. The debriefing will include: name of project, criteria for project, area of growth during project, one success during the project, and one struggle and how it was overcome.

Logbook Entry – Students will record debriefing information in journal and submit journals for teacher review.

Day 24 **Class Activity** – Introduction to Ozobots®

Instruction –Introduce Ozobots® through short video. Explain the care and use of Ozobot® equipment and iPads. Students will begin with tutorials.

Resource(s) – Ozoblocky® introductory video: <http://ozoblockly.com/>

Logbook Entry – Record notes

Day 25 **Class Activity** – Students will work in pairs with the Ozobot® Bits to discover how they function.

Instruction – Students will work in pairs with one Ozobot® Bit and iPad to get acclimated to them. Start with drawing a line on the iPad and having the Bit follow the line. Change the color of the line and then move into adding functions for turning, looping and jumping.

Logbook Entry – Record notes

Day 26 **Class Activity** – Students will work through a series of lessons provided in the teacher packet. These investigations will take 4 days.

Instruction – Ozobot® Teacher lesson plans. Students will learn how Ozobot® senses its environment to move and give commands using OzoCodes (color code language).

Resource(s) – Teacher lesson plans from Ozobot® teacher packet

Logbook Entry – Record findings and include questions, ideas, and concerns.

Day 27 **Class Activity** – Review content from previous lesson. Students will learn how to control Ozobot® with directional OzoCode commands

Instruction – Students will learn how to control Ozobot® through a maze and OzoCode commands. Students will put OzoCode commands on the line such as turn left, turn right, etc. for the Ozobot® to follow. Lesson 3 teaches students how to use static and flash OzoCodes.

Resource(s) – Teacher lesson plan Basic Training Lesson 2 & 3

Logbook Entry – Record OzoCode commands

Day 28 **Class Activity** – Review content from previous lesson. Students will be introduced to computer science concept of programs and functions. OzoGroove is a free app that allows students to write, debug and run a program of several functions.

Instruction – Students will use the free app, OzoGroove®, to write, debug, and run a program with several functions. The Ozobot® will move in a path that forms the letters OZO.

Resource(s) – Ozobot® teacher lesson plan: OzoGroove®

Logbook Entry – Record commands and findings

Day 29 **Class Activity** – Students will design an OzoBlockly® program.

Instruction – Review content from previous lesson. The program that students create contains a growing pattern. Students should be able to explain how they used a growing pattern within the course they created for the Ozobot®. Differentiation can be represented by the complexity of the pattern created.

Resource(s) – Ozobot® teacher Lesson plan: Growing Patterns

Logbook Entry – Record commands and notes

Day 30 **Class Activity** – Review content from all lessons. Ozobot® Bit Second Timer (2-3 day lesson).

Instruction – Students will turn Ozobot® Bit into a timer for measuring seconds. This lesson involves knowledge of concepts from geometry and physical science.

Resource – Stopwatch or phone stopwatch app is required for this lesson.

Logbook Entry – Record commands and notes

Day 32 **Class Activity** – Review the content learned since working with Ozobots®. Introduce the Ozobot® project.

Instruction – Introduce project and rubric. Review content learned from beginning of section. Inform students that information learned in previous lessons will help them create code for the final project in the class.

Resource(s) – Logbook

Logbook Entry – Record notes, questions and concerns

Day 33 **Class Activity** – Students will design the Ozobot® project using OzobotBlockly® graphical drop and drag language.

Instruction – Students will begin researching ideas for their project and recording information in the logbooks. Then students will go through the design process as in the previous lesson. Students will be working in pairs on individual projects. The next 7 days will be workdays with the instructor circulating through the room to answer questions and prompt thinking.

Resource(s) – Logbook and rubric

Logbook Entry – Record information for project, questions and concerns.

Day 40 **Class Activity** – Test and debug program.

Instruction – Students will test their programs with a partner to share results and look for ways to debug software. Time left will be spent rewriting code.

Resource(s) – Logbook

Logbook Entry – Record test results and coding bugs and then changes.

Day 41 **Class Activity** – Independent workday

Instruction – Students will work independently on programs based on needs from testing.

Resource(s) – Logbook

Logbook Entry – Record any errors and changes to errors

Day 42 **Class Activity** – Students will reflect on process of Ozobot® project and take a post-test to show mastery of content.

Instruction – Students will reflect in their logbooks on the process of creating the project. The debriefing will include: name of project, criteria for project, area of growth during project, one struggle and how the student overcame the “struggle” and one success during the project.

Resource(s) – Logbook, post-test

Logbook Entry – The reflection will be in the logbook for teacher review.