

COURSE TITLE

Coding 101

LENGTH

One Semester

DEPARTMENT

Computer Technology
Barbara O'Donnell, Supervisor

SCHOOL

Union Middle School

DATE

Fall 2016

Coding 101

I. Introduction/Overview/Philosophy

In this course, students will explore a variety of software programming languages and coding concepts. The students will use coding to develop games, digital stories, and other interactive designs. Students will be introduced to coding apps, graphics, games, and websites and will design, test, and refine their creations. Digital literacy will also be reviewed with an emphasis on the recognition and prevention of cyberbullying.

II. Objectives

Students will:

- Create interactive scenes with actors, scenes and sound demonstrating an understanding of events and interactions
- Design animations using loops
- Program motion along x- and y-axes
- Build algorithms using conditional logic
- Understand local and global variables, functions, and object cloning
- Understand scripts running in parallel
- Use advanced conditional logic with math and Boolean operators
- Create different scenarios and effects in games
- Troubleshoot and debug simple programs
- Publish projects

III. Course Outline

- 1) Define computer programming
 - a) Understand the definition of computer science programming
 - b) Introduce problem solving/algorithms/computational thinking
- 2) Drawing Basics
 - a) Make drawings with code
 - b) Draw shapes with code
- 3) Coloring
 - a) Color with code
 - b) Outline shapes
- 4) Variables
 - a) Introduce variables
 - b) Use variables to hold values
 - c) Use list variables
 - d) Use persistent cloud variables

- 5) Animation Basics
 - a) Understand animations
 - b) Design animations
- 6) Functions
 - a) Understand function parameters
 - b) Use function return values
 - c) Use local and global functions
- 7) Logic
 - a) Understand booleans
 - b) Use logical operators
 - c) Use If statements
 - d) Use If/Else statements
- 8) Debug Programs
- 9) Looping
 - a) Use while loops
 - b) Use nested loops
 - c) Use conditional loops
- 10) Arrays
 - a) Understand arrays
 - b) Loop through Arrays
- 11) Career Exploration
 - a) Research and explore a variety of computer science related careers

New Jersey Student Learning Standards

TECHNOLOGY

Standard 8.1: Educational Technology: All students will use digital tools to access, manage, evaluate, and synthesize information in order to solve problems individually and collaboratively and to create and communicate knowledge.

Standard 8.2: Technology Education, Engineering, and Design: All students will develop an understanding of the nature and impact of technology, engineering, technological design, and the designed world, as they relate to the individual, global society, and the environment.

Strand E. Computational Thinking: Programming: Computational thinking builds and enhances problem solving, allowing students to move beyond using knowledge to creating knowledge.

21ST CENTURY LIFE AND CAREERS

Standard 9.2: Career Awareness, Exploration, and Preparation

Standard 9.3 – Career & Technical Education (CTE)

Pathway: Programming & Software Development (IT□PRG)

IV. Proficiency Levels

This course is open to grades 7 and 8.

V. Methods of Assessment

Student Assessment

The teacher will provide a variety of assessments during the course of the year. Among these are: homework, laboratory exercises, weekly projects, teacher-made tests and quizzes, and long-term projects.

Curriculum/Teacher Assessment

The teacher will provide the subject area supervisor with suggestions for changes on an ongoing basis.

VI. Grouping

This is a middle school elective course offered to students in grade 7 and grade 8. There is no prerequisite for this course.

VII. Articulation/Scope & Sequence/Time Frame

Course length is one semester and is offered to students in grades 7 and 8.

VIII. Resources

Resources include but are not limited to:

- Khan Academy: Intro to JavaScript <https://www.khanacademy.org/computing/computer-programming/programming>
- <https://code.org/> Accelerated Intro to CS Course including unplugged activities
- <https://www.codeacademy.com/>
- <https://www.codeschool.com/>
- <http://www.inc.com/larry-kim/7-more-places-to-learn-to-code-for-free.html>

IX. Methodologies

The following methods of instruction are suggested: lecture, group projects, demonstration, hands-on applications, and class presentations.

X. Suggested Activities

- Laboratory programming problems

- Game simulated programs
- Cooperative programming projects

XI. Interdisciplinary Connections

Connections are made to mathematics by using a variety of arithmetic formulas. Connections are also made to the disciplines of business, art and English, by means of incorporation of these ideas into programming projects.

XII. Differentiating Instruction for Students with Special Needs: Students with Disabilities, English Language Learners, and Gifted & Talented Students

Differentiating instruction is a flexible process that includes the planning and design of instruction, how that instruction is delivered, and how student progress is measured. Teachers recognize that students can learn in multiple ways as they celebrate students' prior knowledge. By providing appropriately challenging learning, teachers can maximize success for all students.

Examples of Strategies and Practices that Support:

Students with Disabilities

- Use of visual and multi-sensory formats
- Use of assisted technology
- Use of prompts
- Modification of content and student products
- Testing accommodations
- Authentic assessments

English Language Learners

- Pre-teaching of vocabulary and concepts
- Visual learning, including graphic organizers
- Use of cognates to increase comprehension
- Teacher modeling
- Pairing students with beginning English language skills with students who have more advanced English language skills
- Scaffolding
 - word walls
 - sentence frames
 - think-pair-share
 - cooperative learning groups

Gifted & Talented Students

- Adjusting the pace of lessons

- Curriculum compacting
- Inquiry-based instruction
- Independent study
- Higher-order thinking skills
- Interest-based content
- Student-driven
- Real-world problems and scenarios

XIV. Professional Development

The teacher will continue to improve expertise through participation in a variety of professional development opportunities.

Curriculum Map

Week 1	Week 2	Week 3	Week 4	Week 5
What is Computer Programming? Introduction to problem solving for programming (i.e., "how to think about solving the problem" including techniques such as pseudo-code or flowcharts)	Understand the difficulty of translating real problems into programs Drawing Basics		Coloring	

Week 6	Week 7	Week 8	Week 9	Week 10
Variables		Animation Basics		Functions

Week 11	Week 12	Week 13	Week 14	Week 15
Functions	Logic		Debugging	

Week 16	Week 17	Week 18	Week 19	Week 20
Looping and Arrays			Final Project	