COURSE TITLE

Computer Programming 1

LENGTH

One Semester
Grades 9-12

DEPARTMENT

Computer Department
Barbara O'Donnell, Supervisor

SCHOOL

Rutherford High School

DATE

Spring 2015

COMPUTER PROGRAMMING 1

I. Introduction/Overview/Philosophy

Computer Programming 1 is a course appropriate for students with a wide range of backgrounds who intend to further their knowledge of computer design and computer programming in upper level high school computer classes, college, business, or vocational schools. Students will learn computer programming in an animated environment using ALICE. ALICE exposes students to object-oriented programming by creating animated movies and simple video games. This course will provide a foundation for more advanced computer programming languages and computer design techniques.

II. Objectives

Course Outline:

I. Getting started with ALICE
   A. Introduction to ALICE
   B. Introduction to Computer Science
   C. Introduction to object-oriented programming with 3D models and virtual worlds
   D. Learn basic concepts used in ALICE

II. Program Design and Implementation
   A. Create visual and textual storyboards
   B. Translate storyboards to program code
   C. Learn syntax of the ALICE code in reference to a program
   D. Implement built-in methods into scene
   E. Orientate objects through movement

III. Creating a Scenario
   A. Learn the four pieces to putting a scenario together
   B. Use built-in functions and Boolean expression to check conditions of world
   C. Make decisions in your world
   D. Use a repetition control structure to repeat a section of the world

IV. Classes, Objects, Methods and Parameters
   A. Create world-level methods and object-level methods
   B. Inclusion of parameters in both methods
   C. Create new classes using object-level methods
   D. Use inheritance to redefine classes
   E. Use step-wise refinement to complete a complex task

V. Interaction: Events and Event Handling
   A. Create events through user input
   B. Link events to event handling methods
   C. Create interactive event-driven worlds through the concept of event-driven programming
   D. Incorporate incremental development as a technique to debug a program

VI. Using Functions and Control Statements
   A. Use abstraction to create programmer-defined functions
   B. Use Boolean conditions to control the flow of a function
   C. Include parameters in functions for use of different objects
   D. Write class-level functions and save it with different objects

Student Outcomes:

Upon completion of the course, students will demonstrate the ability to:

- Develop an intuition for syntax in program creation
- Use methods, functions, variables, parameters and events to create on-screen movies and games
- Analyze and solve problems by using logical reasoning
- Code the ALICE language fluently in a well-structured fashion
- Communicate complex ideas simply and decompose problems logically
- Use a visual environment to support an objects-early approach as described in the ACM and IEEE-CS Computing Curricula

New Jersey Core Curriculum Content 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)

COMMON CORE STANDARDS LANGUAGE ARTS LITERACY

CCSS.ELA-Literacy.W.11-12.1 Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.

CCSS.ELA-Literacy.W.11-12.2 Write informative/explanatory texts to examine and convey complex ideas, concepts, and information clearly and accurately through the effective selection, organization, and analysis of content.

CCSS.ELA-Literacy.W.11-12.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

CCSS.ELA-Literacy.SL.11-12.5 Make strategic use of digital media (e.g., textual, graphical, audio, visual, and interactive elements) in presentations to enhance understanding of findings, reasoning, and evidence and to add interest.

CCSS.ELA-Literacy.L.11-12.1 Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.

CCSS.ELA-Literacy.L.11-12.2 Demonstrate command of the conventions of standard English capitalization, punctuation, and spelling when writing.

COMMON CORE STANDARDS MATHEMATICS

High School—Algebra HSA-SSE: **Seeing Structure in Expressions** Interpret parts of an expression; Use the structure of an expression to identify ways to rewrite it.

III. Proficiency Levels

This course is open to grades 9 -12.

IV. 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.

V. Grouping

There is no prerequisite for Computer Programming 1. Students completing this course will have sufficient knowledge and experience with data types and structures to be successful in advanced programming and design courses such as C++, Java, and Multimedia Design.

VI. Articulation/Scope & Sequence/Time Frame

Course length is one semester and is offered to students in grades 9-12.

VII. Resources

Learning to Program with ALICE. Wanda Dann, Stephen Cooper, Randy Pausch. Prentice Hall, 2012.

Starting out with Alice, A visual Introduction to Programming. Tony Gaddis. Pearson Education, Inc., 2013.

Alice in action with Java. Joel Adams. Calvin College, 2008

Companion website: http://www.alice.org

VIII. Methodologies

Much of the class time is spent in lab work with required hands-on exercises to be completed. When group instruction is necessary, topics are taught using the computer projection system, in conjunction with student classwork.

IX. Suggested Activities

- Laboratory programming problems
- Game simulated programs
- Cooperative programming projects

X. Interdisciplinary Connections

Connections are made to mathematics by using a variety of arithmetic formulas, as well as higher mathematical concepts. Connections are also made to the disciplines of business, art and English, by means of incorporation of theses ideas into programming projects.

XI. 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

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

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

XII. Professional Development

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

MONTH	CONTENT	SKILLS	ASSESSMENT
SEPTEMBER / FEBRUARY	• What is Alice? • What is Computer Science? • What is a computer program? • What is an Object? • Alice 3D Library • Animations • Design and implement a program - storyboard • Algorithm • Do-In Order • Do-together • Object properties • Vehicles • Comments • Simple control structures • Expressions • Built in functions	• Differentiate computer science from computer programming • Design an Alice program • Use the Alice 3D Library • Write a simple Alice program • Make objects interact with each other • Design a program with simple control structures • Use the built in functions and expressions of each object	• Alice Scavenger Hunt • Questions from the reading • Lab assignments • Quizzes – multiple choice and free response
OCTOBER / MARCH	• Classes • Object Oriented • World-Level Methods • Class Level Methods • Sound • Computer Ethics	• Identify and use classes. • Calling built in methods • Designing and writing own methods • Use and create parameters • Create new classes with own methods • Identify and use class level vs. world level methods • Identify computer ethics	• Questions from the reading • Programming exercises • Lab assignments • Quizzes – multiple choice and free response • Project # 1 – Greeting Card
NOVEMBER / APRIL	• Event Handling • Functions • Control Statements • Random numbers	• Identify possible uses of event-driven programming • Read and write if/else control statements • Write programs that respond to the keyboard • Design and create an interactive world using built in event handling methods and control statements	• Questions from the reading • Programming exercises • Lab assignments • Quizzes – multiple choice and free response • Project # 2 – Interactive World

MONTH	CONTENT	SKILLS	ASSESSMENT
DECEMBER / MAY	• Definite and conditional loops • Variables • Lists • STEM careers	• Write and trace code in definitive loops • Write and follow code in conditional loops • Determine types of loops to use • Identify STEM careers If time permits • Create and use variables • Create and use lists	• Questions from the reading • Programming exercises • Lab assignments • Quizzes – multiple choice and free response
JANUARY / JUNE	• Create an interactive Game using the skills learned during this class	• Integrate the skills learned over the year to create an interactive animation • Present the final program to the class Design a game using storyboarding • Create the game using Alice • Present the game to the class	• Project #3 – Game including design, creation and presentation • Final exam