

Wilson Area School District Planned Course Guide

Title of planned course: Introduction to Computer Programming

Subject Area: Business

Grade Level: 9-12

Course Description: In this course, students are introduced to basic computer programming concepts through the Alice system. Alice is a software program that teaches object-oriented computing in a 3D environment. The learning objectives and core activities prepare students to read and write programs in the java language. The skills developed in this course are essential because Java is the standard language for the World Wide Web. The topics covered include: the evolution of computer programming, program design, methods, variables, events, lists, arrays, and others. Students will complete a final project at the conclusion of the course that demonstrates satisfactory application of the stated learning objectives.

Time/Credit for this Course: 1 credit

Curriculum Writing Committee: Daryl Feldman

Curriculum Map

August:

- Introduction to Programming (Unit I)

September:

- Introduction to Programming – continued (Unit I)
- Getting Started with Alice (Unit II)
- Program Design and Implementation (Unit III)

October:

- Program Design and Implementation – continued (Unit III)
- Methods (Unit IV)

November:

- Variables, Functions, Math Expressions, and Strings (Unit V)

December:

- Classes and Parameters (Unit VI)

January:

- Interaction: Events and Event Handling (Unit VII)

February:

- Decision and Repetition Structures (Unit VIII)

March:

- Lists and Arrays (Unit IX)

April:

- Introduction to Java (Unit X)

May:

- Final Project: Interactive Program (video game)

June:

- Final Project: Interactive Program (video game) Presentations

**Wilson Area School District
Planned Course Materials**

Course Title: Introduction to Computer Programming

Textbook: Learning to Program with Alice: Second Edition
Pearson Education, 2009

Supplemental Books: Starting Out with Alice: A Visual Introduction to
Programming
Pearson Education, 2008

Teacher Resources: www.alice.org
www.aliceprogramming.net
<http://www.cs.duke.edu/csed/alice/aliceInSchools/>
<http://www.dickbaldwin.com/tocalice.htm>
<http://www.alice.org/community/>
www.bluepelicanjava.com

Curriculum Scope & Sequence

Planned Course: Introduction to Computer Programming

Unit: Introduction to Programming (Unit I)

Time frame: 2 weeks

State Standards: 3.4.10.B4, 3.4.10.C2

Anchor(s) or adopted anchor: S11.A.2.2

Essential content/objectives: At end of the unit, students will be able to:

1. define computer program
2. summarize the history of computer programming
3. identify current challenges associated with the computer programming field
4. describe the benefits of using Alice to teach computer programming fundamentals (object-oriented programming)

Core Activities: Students will complete/participate in the following:

- assignment: essential terms
- activity: locker combination instructions (algorithm example)
- WebQuest: the evolution of computer programming
- written report: challenges facing computer programming industry today
- assignment: Alice K-W-L chart

Extensions:

- assignment: evolution of computer programming timeline
- activity: importance of computer programming in today's society
- guest speaker: an individual who is employed in the computer programming industry
- critical thinking / problem solving opportunities

Remediation:

- ABC Taxonomy: "I know, I know something about...computer programming."
- guided reading assignments
- review questions / worksheets
- tutoring
- retests

Instructional Methods: direct instruction, guided practice, independent exercises, reading for understanding, cooperative learning – "think, pair, share", large and small group discussion, reflective writing, visual aids / graphic organizers, valid Internet research, technology assisted learning

Board Approved October 2011

Materials & Resources: Learning to Program with Alice textbook, computer, Internet, SMART Board, projector, Alice promotional video, The Last Lecture, Microsoft Office programs

Assessments: questioning, quiz, homework, test

Curriculum Scope & Sequence

Planned Course: Introduction to Computer Programming

Unit: Getting Started with Alice (Unit II)

Time frame: 1-2 weeks

State Standards: 3.4.12.A3, 3.4.12.C2, 3.4.12.D2

Anchor(s) or adopted anchor: S11.A.2.1

Essential content/objectives: At end of the unit, students will be able to:

1. define computer programming basic terminology (algorithm, text editor, flowchart)
2. describe basic Alice concepts (virtual world, 3D model, directions, orientation, distance, position, and animation)
3. understand the relationship between objects and classes
4. use the Alice environment (templates, code editor, event editor, scene editor, gallery, and quad view)
5. change an object's properties
6. manipulate an object's subparts
7. control the camera (up, down, left, right, forward, backward, rotation, and tilt)
8. construct a virtual world containing text and 2D images (billboards)

Core Activities: Students will complete/participate in the following:

- assignment: essential terms
- assignment: Alice scavenger hunt
- activity: Alice "Simon says"
- tutorial (step-by-step): opening and playing an Alice world
- tutorial (step-by-step): creating a world and adding objects
- tutorial (step-by-step): moving the camera in 3D space
- tutorial (step-by-step): manipulating objects in 3D space

Extensions:

- setting up an initial scene (to be used in first program)
- student presentation: initial scene set up
- critical thinking / problem solving opportunities

Remediation:

- guided reading assignments
- review questions / worksheets
- textbook exercises in support of learning objectives
- tutoring
- retests

Instructional Methods: direct instruction, guided practice, independent exercises, reading for understanding, cooperative learning – “think, pair, share”, large and small group discussion, reflective writing, visual aids / graphic organizers, student presentations, games, technology assisted learning

Materials & Resources: Learning to Program with Alice textbook, Alice software, computer, Internet, SMART Board, projector, PowerPoint files

Assessments: questioning, quiz, homework, test

Curriculum Scope & Sequence

Planned Course: Introduction to Computer Programming

Unit: Program Design and Implementation (Unit III)

Time frame: 2-3 weeks

State Standards: 3.4.12.B1, 3.4.12.C2, 3.4.12.D2

Anchor(s) or adopted anchor: S11.A.2.1, S11.A.3.1

Essential content/objectives: At end of the unit, students will be able to:

1. define scenario and storyboard
2. differentiate between textual and visual storyboards
3. read a scenario and create a storyboard as a guide for writing a program
4. describe the four-step programming process and the problem solving techniques used
5. write a scenario and design a flowchart for program implementation
6. translate a storyboard to program code
7. demonstrate proper use of control statements (Do in order, Do together)
8. select arguments that support the intended program actions
9. export code for printing

Core Activities: Students will complete/participate in the following:

- assignment: essential terms
- tutorial (step-by-step): adding instructions to an Alice world
- tutorial (step-by-step): exploring primitive methods
- tutorial (step-by-step): using the program design cycle
- program development: recreate the first program (robot in space)
- project – program design: create an original scenario, storyboard, and flow chart
- program development: write first original program (short animated clip)

Extensions:

- create a graphic organizer that displays the design and implementation stages of program development
- translate classmates' storyboards to program code
- student presentations: share first original program with class
- submit program code to teacher via email
- critical thinking / problem solving opportunities

Remediation:

- guided reading assignments
- review questions / worksheets
- textbook exercises in support of learning objectives
- flashcards (essential term review)

- tutoring
- retests

Instructional Methods: direct instruction, guided practice, independent exercises, reading for understanding, cooperative learning – “think, pair, share”, large and small group discussion, reflective writing, visual aids / graphic organizers, student presentations, games, technology assisted learning, project-based learning with performance standards

Materials & Resources: Learning to Program with Alice textbook, Alice software, computer, Internet, SMART Board, projector, PowerPoint files, Microsoft Office programs, printer

Assessments: questioning, quiz, programming exercises, homework, peer review, projects, test, self-reflection

Curriculum Scope & Sequence

Planned Course: Introduction to Computer Programming

Unit: Methods (Unit IV)

Time frame: 2-3 weeks

State Standards: 3.4.10.D2, 3.4.12.C3

Anchor(s) or adopted anchor: S11.A.3.1, S11.A.3.3

Essential content/objectives: At end of the unit, students will be able to:

1. identify primitive methods (built-in)
2. explain the relationship between method calls and arguments
3. write their own methods
4. recall standard conventions for naming objects and methods
5. label essential pieces of program code with comments
6. modify the properties of an object
7. create and rename new objects from predefined classes
8. break down a program problem through testing and debugging

Core Activities: Students will complete/participate in the following:

- assignment: essential terms
- tutorial (step-by-step): inserting comments
- tutorial (step-by-step): creating simultaneously executed instructions
- activity: debugging - what is wrong with this program?
- project: write methods to recreate a scene from your favorite movie

Extensions:

- role play: students act out classmates' program code
- contest (race): modifying the properties of an object
- create a graphic organizer that lists the standard conventions for naming objects and methods
- critical thinking / problem solving opportunities

Remediation:

- guided reading assignments
- review questions / worksheets
- textbook exercises in support of learning objectives
- flashcards (essential terms review)
- review game
- tutoring
- retests

Instructional Methods: direct instruction, guided practice, independent exercises, reading for understanding, cooperative learning – “think, pair, share”, large and small group discussion, reflective writing, visual aids / graphic

organizers, student role play, contests, games, technology assisted learning, project-based learning with performance standards

Materials & Resources: Learning to Program with Alice textbook, Alice software, computer, Internet, SMART Board, projector, PowerPoint files, Microsoft Office programs, printer

Assessments: questioning, quiz, programming exercises, homework, peer review, projects, test, self-reflection

Curriculum Scope & Sequence

Planned Course: Introduction to Computer Programming

Unit: Variables, Functions, Math Expressions, and Strings (Unit V)

Time frame: 3-4 weeks

State Standards: 3.4.12.A3, 3.4.12.C3, 2.2.11.C, 2.5.11.A, 2.5.11.B, 2.5.11.E, 2.8.11.E

Anchor(s) or adopted anchor: S11.A.3.3, M11.A.1, M11.A.2, M11.A.3, M11.B.2, M11.D.1.1

Essential content/objectives: At end of the unit, students will be able to:

1. create a variable (local variable and set instruction)
2. select appropriate variable types (number, Boolean, object, string, color, sound, etc.)
3. use variables in expressions to calculate new results
4. understand the difference between methods and functions
5. include functions in programs (built-in and ask the user)
6. solve problems by using functions
7. identify math operators
8. use math expressions to move objects while avoiding collisions
9. produce strings and 3D text
10. implement visual effects and animation (change the point of view by creating dummy objects, capture poses, make an object circle another object)

Core Activities: Students will complete/participate in the following:

- assignment: essential terms
- tutorial (step-by-step): creating and using a variable
- tutorial (step-by-step): creating a set instruction for a variable
- tutorial (step-by-step): calling an ask user function
- tutorial (step-by-step): using a proximity function
- tutorial (step-by-step): using math to avoid collisions
- tutorial (step-by-step): converting a number variable to a string
- project: phishy move (functions)

Extensions:

- program development: write a short program that utilizes built-in functions
- project: create a program that teaches a basic math lesson (i.e.: miles per gallon)
- mid-course movie animation (ongoing throughout first semester)
- student demonstrations: how to....(as related to tutorials)
- critical thinking / problem solving opportunities

Remediation:

- guided reading assignments

- review questions / worksheets
- textbook exercises in support of learning objectives
- flashcards (essential terms review)
- review game
- tutoring
- retests

Instructional Methods: direct instruction, guided practice, independent exercises, reading for understanding, cooperative learning – “think, pair, share”, large and small group discussion, reflective writing, visual aids, student demonstrations, games, technology assisted learning, project-based learning with performance standards

Materials & Resources: Learning to Program with Alice textbook, Alice software, computer, Internet, SMART Board, projector, PowerPoint files, Microsoft Office programs

Assessments: questioning, quiz, programming exercises, homework, peer review, projects, test, self-reflection

Curriculum Scope & Sequence

Planned Course: Introduction to Computer Programming

Unit: Classes and Parameters (Unit VI)

Time frame: 2-3 weeks

State Standards: 3.4.10.D2, 2.2.11.C

Anchor(s) or adopted anchor: S11.A.3.3, M11.A.2, M11.A.3

Essential content/objectives: At end of the unit, students will be able to:

1. understand abstraction
2. explain the purpose of using parameters in writing programs
3. break down a program into smaller pieces by using objects and writing their own methods (stepwise refinement)
4. create parameters (number, Boolean, object, string, color, sound, etc.)
5. design methods to accept parameters
6. describe inheritance
7. import sound files
8. create a new class
9. save an object to a new class
10. understand the need for writing your own function
11. write class-level functions

Core Activities: Students will complete/participate in the following:

- assignment: essential terms
- tutorial (step-by-step): creating and placing parameters
- assignment: revise a storyboard to accommodate parameters
- activity: search for sound files, download sound files, import sound files
- tutorial (step-by-step): creating and saving a new class

Extensions:

- graphic organizer: Venn diagram – abstraction v. stepwise refinement
- activity: testing parameters with arguments
- program development: write program code using multiple parameters
- project: writing class level methods and creating a new class
- student demonstrations: how to....(as related to tutorials)
- critical thinking / problem solving opportunities
- mid-course movie animation (ongoing throughout first semester)

Remediation:

- guided reading assignments
- review questions / worksheets
- textbook exercises in support of learning objectives
- flashcards (essential terms review)

- review game
- tutoring
- retests

Instructional Methods: direct instruction, guided practice, independent exercises, reading for understanding, cooperative learning – “think, pair, share”, large and small group discussion, reflective writing, visual aids, student demonstrations, games, technology assisted learning, project-based learning with performance standards

Materials & Resources: Learning to Program with Alice textbook, Alice software, computer, Internet, SMART Board, projector, PowerPoint files, Microsoft Office programs

Assessments: questioning, quiz, programming exercises, homework, peer review, projects, test, self-reflection

Curriculum Scope & Sequence

Planned Course: Introduction to Computer Programming

Unit: Interaction: Events and Event Handling (Unit VII)

Time frame: 3 weeks

State Standards: 3.4.12.C2, 3.4.12.D2, 2.2.11.C

Anchor(s) or adopted anchor: S11.A.2.1, M11.A.2, M11.A.3

Essential content/objectives: At end of the unit, students will be able to:

1. define events
2. create events to respond to user input
3. differentiate between event types
4. identify the events that fire once and those that fire repeatedly
5. use events to monitor conditions and variable changes
6. understand the BDE event
7. explain how events make programs interactive
8. implement events to build sophisticated program code

Core Activities: Students will complete/participate in the following:

- assignment: essential terms
- tutorial (step-by-step): handling key press events
- tutorial (step-by-step): handling the while a key is pressed event
- tutorial (step-by-step): handling a mouse click event
- activity: create a short program that includes a BDE event

Extensions:

- graphic organizer: event types
- assignment: storyboard design with events
- activity: understanding incremental development
- project: students will demonstrate the ability to use events in programs
- student demonstrations: how to....(as related to tutorials)
- critical thinking / problem solving opportunities

Remediation:

- guided reading assignments
- review questions / worksheets
- textbook exercises in support of learning objectives
- flashcards (essential terms review)
- review game
- tutoring
- retests

Instructional Methods: direct instruction, guided practice, independent exercises, reading for understanding, cooperative learning – “think, pair, share”,

large and small group discussion, reflective writing, visual aids, student demonstrations, games, technology assisted learning, project-based learning with performance standards

Materials & Resources: Learning to Program with Alice textbook, Alice software, computer, Internet, SMART Board, projector, PowerPoint files, Microsoft Office programs

Assessments: questioning, quiz, programming exercises, homework, peer review, projects, test, self-reflection

Curriculum Scope & Sequence

Planned Course: Introduction to Computer Programming

Unit: Decision and Repetition Structures (Unit VIII)

Time frame: 4-5 weeks

State Standards: 2.4.11.B, 2.5.11.A, 2.5.11.B, 2.8.11.E

Anchor(s) or adopted anchor: M11.A.1, M11.A.2, M11.D.1.1

Essential content/objectives: At end of the unit, students will be able to:

1. write if/else statements
2. use relational operators for condition based decisions
3. determine Boolean results
4. demonstrate knowledge of logical operators to create complex situations
5. create nested if/else statements and loops
6. utilize while and loop statements
7. use control loop variable
8. recognize proper decision control statements
9. draw a flowchart for a decision structure
10. describe how to use recursion in a program

Core Activities: Students will complete/participate in the following:

- assignment: essential terms
- tutorial (step-by-step): creating an if/else instruction
- tutorial (step-by-step): using a relational operator
- tutorial (step-by-step): using a loop instruction
- tutorial (step-by-step): using the while instruction
- project: pyramid climb
- tutorial (step-by-step): creating a recursive method
- tutorial (step-by-step): recursive problem solving in animation
- tutorial (step-by-step): writing a recursive mathematical function

Extensions:

- assignment: flowchart with a single alternative decision structure
- assignment: flowchart with a nested decision structure
- assignment: flowchart of a loop / while instruction
- project: driving test
- student demonstrations: how to....(as related to tutorials)
- critical thinking / problem solving opportunities

Remediation:

- guided reading assignments
- review questions / worksheets
- textbook exercises in support of learning objectives

- flashcards (essential terms review)
- review game
- tutoring
- retests

Instructional Methods: direct instruction, guided practice, independent exercises, reading for understanding, cooperative learning – “think, pair, share”, large and small group discussion, reflective writing, visual aids, student demonstrations, games, technology assisted learning, project-based learning with performance standards

Materials & Resources: Learning to Program with Alice textbook, Alice software, computer, Internet, SMART Board, projector, PowerPoint files, Microsoft Office programs

Assessments: questioning, quiz, programming exercises, homework, peer review, projects, test, self-reflection

Curriculum Scope & Sequence

Planned Course: Introduction to Computer Programming

Unit: Lists and Arrays (Unit IX)

Time frame: 3-4 weeks

State Standards: 3.4.12.C3,

Anchor(s) or adopted anchor: S11.A.2.1

Essential content/objectives: At end of the unit, students will be able to:

1. create lists and arrays
2. modify lists and arrays
3. compare and contrast lists and arrays
4. use an array visualization object
5. develop list methods and functions
6. access elements stored in lists and arrays
7. change the contents of lists
8. randomly select an object in a list or array

Core Activities: Students will complete/participate in the following:

- assignment: essential terms
- tutorial (step-by-step): creating a list and using for all in order and for all together instructions
- tutorial (step-by-step): more complex list processing
- tutorial (step-by-step): using the let the mouse move objects event
- tutorial (step-by-step): creating an array and a loop that steps through it
- tutorial (step-by-step): randomly selecting an array element

Extensions:

- activity: dominoes
- program development: bug zapper game
- project: whack-a-mole
- game development: frog crossing the pool
- student demonstrations: how to....(as related to tutorials)
- critical thinking / problem solving opportunities

Remediation:

- guided reading assignments
- review questions / worksheets
- textbook exercises in support of learning objectives
- flashcards (essential terms review)
- review game
- tutoring
- retests

Instructional Methods: direct instruction, guided practice, independent exercises, reading for understanding, cooperative learning – “think, pair, share”, large and small group discussion, reflective writing, visual aids, student demonstrations, games, technology assisted learning, project-based learning with performance standards

Materials & Resources: Learning to Program with Alice textbook, Alice software, computer, Internet, SMART Board, projector, PowerPoint files, Microsoft Office programs

Assessments: questioning, quiz, programming exercises, homework, peer review, projects, test, self-reflection

Curriculum Scope & Sequence

Planned Course: Introduction to Computer Programming

Unit: Introduction to Java (Unit X)

Time frame: 4 weeks

State Standards: 3.2.12.D, 3.4.10.E4, 3.7.12.A, 3.7.12.C, 2.5.11.B

Anchor(s) or adopted anchor: S11.A.2.1, S11.A.3.3, M11.A.2

Essential content/objectives: At end of the unit, students will be able to:

1. compare and contrast the Alice and Java programming environments
2. explain how source code is translated to byte code
3. access Java API support library and online documentation
4. explain what actions a code written in Java will perform
5. understand why syntax errors occur
6. examine the reasons bugs occur in Java programs
7. use a constructor to create new objects
8. create Java statements

Core Activities: Students will complete/participate in the following:

- assignment: essential terms
- assignment: edit pre-written java programs
- assignment: debug pre-written java programs

Extensions:

- activity: write program statements using java language
- graphic organizer: Venn diagram – Alice v. Java
- critical thinking / problem solving opportunities

Remediation:

- guided reading assignments
- review questions / worksheets
- textbook exercises in support of learning objectives
- tutoring
- retests

Instructional Methods: direct instruction, guided practice, independent exercises, reading for understanding, cooperative learning – “think, pair, share”, large and small group discussion, reflective writing, visual aids, technology assisted learning

Materials & Resources: computer, Internet, SMART Board, projector, PowerPoint files, Microsoft Office programs, Net beans

Assessments: questioning, quiz, programming exercises, homework, peer review, test, self-reflection