Wayne

Township Public Schools

Mathematics
Grades 10th -12th
Honors Computer Programming

Dr. Mark Toback, Superintendent

Committee: Thomas Grasso

This curriculum may be modified through varying techniques, strategies, and materials as per an individual student's Individualized Educational Plan (IEP)

Wayne School District

Curriculum Format

Content Area/ Grade Level/ Course:	**Mathematics 10-12 Honors Programming**
Unit Plan Title:	**Unit 1. Programming Fundamentals - Vars, Data, Using Objects**
Time Frame	**6 Weeks**

Anchor Standards/Domain*   *i.e: ELA: reading, writing i.e.: Math: Number and Operations in Base 10

Data and Analysis	DA
Algorithms and Programming	AP
Engineering Design	ED
21st Century Life Skills: Career Preparation	9.2.12.C
Career & Technical Education: Information Technology	9.3.IT-PRG

Unit Summary

Unit 1. Programming Fundamentals - Vars, Data, Using Objects

A. What is data?
B. What are variables?
C. Primitive data types
D. Open ended Problem Solving
E. Troubleshooting and error correction
F. Design solutions using code

Standard Number(s) i.e: Math: 3.NBT.1   i.e.: RL 8.1

8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.
8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
8.1.12.DA.3: Translate between decimal numbers and binary numbers.
8.1.12.DA.4: Explain the relationship between binary numbers and the storage and use of data in a computing device.
8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects
8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.2.12.ED.2: Create scaled engineering drawings for a new product or system and make modification to increase optimization based on feedback.
9.2.12.CAP.6: Identify transferable skills in career choices and design alternative career plans based on those skills.
9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a)
9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).
WHST.11-12.4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
WHST.11-12.5. Develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach, focusing on addressing what is most significant for a specific purpose and audience.
WHST.11-12.6. Use technology, including the Internet, to produce, share, and update writing products in response to ongoing feedback, including new arguments or information.
NJSLSA.SL1. - Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.
SL.11-12.4 - Present information, findings and supporting evidence clearly, concisely, and logically. The content, organization, development, and style are appropriate to task, purpose, and audience.
HS-ETS1-2. - Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
HS-ETS1-4. - Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Essential Question(s)

- What types of data do we need to model in our programs?
- What are programming conventions and why are they useful?
- What are the positive and negative impacts of technology on society?
- What is programming?
- How should our programs communicate to our end users?
- What does it mean to program in an Object Oriented way?

Enduring Understandings

- Technology and programming languages are constantly evolving.
- Technology has impacted virtually every industry and field of endeavor in our society.
- Programs are fundamentally very specific and literal instructions with specific syntax.
- Object Oriented Programming allows us to create modular, portable, re-usable code in a very powerful way.

In this unit plan, the following 21st Century themes and skills are addressed.

Check all that apply. **21st Century Themes**		*Indicate whether these skills are **E**-Encouraged, **T**-Taught, or **A**-Assessed in this unit by marking **E, T, A** on the line before the appropriate skill.* **21st Century Skills**
X	**Global Awareness**	**E,T,A**
	Environmental Literacy	**E,T,A**
	Health Literacy	**E,T,A**
	Civic Literacy	**E,T,A**
X	**Financial, Economic, Business, and Entrepreneurial Literacy**	

Student Learning Targets/Objectives (Students will know/Students will understand)

- Develop methods for creating possible solutions, modeling and testing solutions, and modifying proposed design in the solution of a technological problem using hands-on activities and computer based programs.
- Create a software based solution to a problem using a given program specification or design document by applying design and software development principles.
- Apply data analysis to the problem-solving and decision making processes in a variety of life situations.
- Engage in an informed discussion about how technology has profoundly impacted various professional career paths.
- Explain the software development lifecycle and why initial planning of data and methods is so critical to the design process.
- Large problems are solved by breaking things down into smaller parts.
- Methods are used to solve specific problems and allow us to organize and optimize our code.
- Verbally explain the process of creating a program, compiling, error-correcting, running and checking solutions based on test cases.
- Identify several uses for computer programming in modern society.
- Explain how objects are created and methods are invoked on these objects.

Assessments (Pre, Formative, Summative, Other) ***Denote required common assessments with an ****

Throughout the course there is a strong emphasis on Object Oriented Programming and related concepts such as Data Encapsu
Students will work from a program specification to build their own creative solutions to problems. By its very nature the course is highly differentiated and each student is free to develop advanced features and enhancements beyond those required for a basic solution. The introduction of GUI (Graphic User Interfaces) early in the course further increases student options for creating exceptional graphics based applets and applications in addition or in place of a console based solution. Backwards Design in the discussion of a program solution will be featured regularly as we carefully consider which type of designs will provide for optimal solutions given the tools that students have acquired to date.

Formative and summative evaluations in the form of worksheets, quizzes and tests - both paper based and computer based will be utilized as needed to reinforce learning objectives and evaluate student progress.

Teaching and Learning Activities

Activities	• Lecture and class discussion. • Video and multimedia presentations. • Build a console, applet and GUI based application. • Review and extend program functionality. • Group and collaborative work. • Student presentations of projects.
Differentiation Strategies	• Individual and collaborative research, design and problem solving • Student interest and skill level assessment • Individual, small group, and large group instruction • Media presentations and guest speakers • Student presentations and Flipped Lessons Differentiation Strategies for Special Education Students Differentiation Strategies for Gifted and Talented Students Differentiation Strategies for ELL Students Differentiation Strategies for At Risk Students

Resources

- **http://www.state.nj.us/education/cccs/**
- **http://www.corestandards.org/ELA-Literacy**
- **http://www.state.nj.us/education/cccs/**

Wayne School District

Curriculum Format

Content Area/ Grade Level/ Course:	**Mathematics 10-12 Honors Programming**
Unit Plan Title:	**Unit 2. Program Flow and Decisions: Loops and Conditionals**
Time Frame	**6 Weeks**

Anchor Standards/Domain*   *i.e: ELA: reading, writing i.e.: Math: Number and Operations in Base 10

Data and Analysis	DA
Algorithms and Programming	AP
Engineering Design	ED
21st Century Life Skills: Career Preparation	9.2.12.C
Career & Technical Education: Information Technology	9.3.IT-PRG

Unit Summary

Unit 2. Program Flow and Decisions: Loops and Conditionals

- What is “Flow of Control”?
- Syntax of “if”, “for”, and “while” loops
- Usage of these conditionals and loops
- Organizing code and blocking statements using curly braces.
- Why is disrupting flow of control needed in programming?
- Problem solving using these statements
- Combining statements to achieve a goal.

Standard Number(s)

8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.
8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects
8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and Accessible
8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.
8.2.12.ED.2: Create scaled engineering drawings for a new product or system and make modification to increase optimization based on feedback.
9.2.12.CAP.6: Identify transferable skills in career choices and design alternative career plans based on those skills.
9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a)
9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).
WHST.11-12.4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
WHST.11-12.5. Develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach, focusing on addressing what is most significant for a specific purpose and audience.
WHST.11-12.6. Use technology, including the Internet, to produce, share, and update writing products in response to ongoing feedback, including new arguments or information.
NJSLSA.SL1. - Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.
SL.11-12.4 - Present information, findings and supporting evidence clearly, concisely, and logically. The content, organization, development, and style are appropriate to task, purpose, and audience.
HS-ETS1-2. - Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

HS-ETS1-4. - Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Essential Question(s)

Unit 2. Program Flow and Decisions: Loops and Conditionals

- How do engineers solve problems using the engineering design process?
- How is creativity and innovation used in engineering design?
- What does it look like to be safe in an engineering lab?
- How do teams efficiently and effectively solve problems in an increasingly complex world?
- What are the elements of good engineering design?

Enduring Understandings

- Flow of control can be disrupted in many useful ways.
- Programmers can use loops to eliminate lines of code and make their programs more efficient.
- Programmers can use conditional statements to skip lines and give the appearance of choice.
- Curly brace placement can drastically affect the function and output of code.
- Neat code is as important as the actual solution.

In this unit plan, the following 21st Century themes and skills are addressed.

	Check all that apply. **21st Century Themes**		*Indicate whether these skills are **E**-Encouraged, **T-**Taught, or **A**-Assessed in this unit by marking **E, T, A** on the line before the appropriate skill.* **21st Century Skills**
X	**Global Awareness**	**E,T,A**	**Creativity and Innovation**
	Environmental Literacy	**E,T,A**	**Critical Thinking and Problem Solving**
	Health Literacy	**E,T,A**	**Communication**
	Civic Literacy	**E,T,A**	**Collaboration**
X	**Financial, Economic, Business, and Entrepreneurial Literacy**		

Student Learning Targets/Objectives (Students will know/Students will understand)

- Develop methods for creating possible solutions, modeling and testing solutions, and modifying proposed design in the solution of a technological problem using hands-on activities.
- How to use loops and conditional statements to solve problems.
- Which type of loop is preferred for different solutions.
- Where to put curly braces to make block statements.
- Formatting of code so other programmers can read their solution easily.
- Apply communications and data analysis to the problem-solving and decision making processes in a variety of life situations.
- Discussion and critique of program development.
- A multitude of smaller systems at work in order for software to operate as designed.

Assessments (Pre, Formative, Summative, Other) ***Denote required common assessments with an ****

Throughout the course there is a strong emphasis on Object Oriented Programming and related concepts such as Data Encapsulation and Information Hiding.
Students will work from a program specification to build their own creative solutions to problems. By its very nature the course is highly differentiated and each student is free to develop advanced features and enhancements beyond those required for a basic solution. The introduction of GUI (Graphic User Interfaces) early in the course further increases student options for creating exceptional graphics based applets and applications in addition or in place of a console based solution. Backwards Design in the discussion of a program solution will be featured regularly as we carefully consider which type of designs will provide for optimal solutions given the tools that students have acquired to date.

Formative and summative evaluations in the form of worksheets, quizzes and tests - both paper based and computer based will be utilized as needed to reinforce learning objectives and evaluate student progress.

Teaching and Learning Activities

Activities	• Lecture and class discussion. • Video and multimedia presentations. • Build a console, applet and GUI based application. • Review and extend program functionality. • Group and collaborative work. • Student presentations of projects.
B *Differentiation Strategies*	• Individual and collaborative research, design and problem solving • Student interest and skill level assessment • Individual, small group, and large group instruction • Media presentations and guest speakers • Student presentations and Flipped Lessons Differentiation Strategies for Special Education Students Differentiation Strategies for Gifted and Talented Students Differentiation Strategies for ELL Students Differentiation Strategies for At Risk Students

Resources
- http://www.state.nj.us/education/cccs/ - http://www.corestandards.org/ELA-Literacy - http://www.state.nj.us/education/cccs/

Wayne School District
Curriculum Format

Content Area/ Grade Level/ Course:	**Mathematics / Computer Science 10-12 Honors Programming**
Unit Plan Title:	**Unit 3. Methods and Objects**
Time Frame	**6 Weeks**
Anchor Standards/Domain* *i.e: ELA: reading, writing i.e.: Math: Number and Operations in Base 10	

Data and Analysis	DA
Algorithms and Programming	AP
Engineering Design	ED
21st Century Life Skills: Career Preparation	9.2.12.C
Career & Technical Education: Information Technology	9.3.IT-PRG

Unit Summary

Unit 3. Methods and Objects

A. Hands-on exploration into programming using methods and objects
B. Build a procedural utility object to solve several mathematical problems.
C. Build several objects to model various geometrical areas and volumes.
D. Practice creating a variety of front-end clients that all utilize the same underlying object to demonstrate the power of reusable code.
E. Design and build both object and client software to solve a variety of practical real world problems and situations.

Standard Number(s)

8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
8.1.12.DA.3: Translate between decimal numbers and binary numbers.
8.1.12.DA.4: Explain the relationship between binary numbers and the storage and use of data in a computing device.8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.
8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects
8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and Accessible
8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.
8.2.12.ED.2: Create scaled engineering drawings for a new product or system and make modification to increase optimization based on feedback.

9.2.12.CAP.6: Identify transferable skills in career choices and design alternative career plans based on those skills.
9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a)
9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).
WHST.11-12.4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
WHST.11-12.5. Develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach, focusing on addressing what is most significant for a specific purpose and audience.
WHST.11-12.6. Use technology, including the Internet, to produce, share, and update writing products in response to ongoing feedback, including new arguments or information.
NJSLSA.SL1. - Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.
SL.11-12.4 - Present information, findings and supporting evidence clearly, concisely, and logically. The content, organization, development, and style are appropriate to task, purpose, and audience.

HS-ETS1-2. - Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
HS-ETS1-4. - Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Essential Question(s)

- Why do we use methods?
- How are methods used to organize code?
- How do we document methods?
- How does a non-static method differ from a static method?
- What purpose do void methods serve?
- How many values can be returned from a method?
- How does an access specifier determine the scope of a method?
- What does it mean to overload a method?
- What constitutes the signature of a method?
- What does it mean to use an object oriented program design?
- How does object oriented programming differ from procedural programming?
- What is the purpose of a constructor?
- What is a default constructor and when does it get invoked?
- How does overloading an object's constructor provide flexibility to client code?
- What is data encapsulation and information hiding?
- How do private data and public methods support data encapsulation?
- How does an object store its state and what constitutes an object's public interface?
- Why are methods the building blocks of objects?

Enduring Understandings

- Methods are used to organize code into clearly defined, reusable blocks.
- Learning to code with methods and objects is a skill that will enhance a programmer's ability to create powerful, modular and reusable code.
- Data encapsulation is a key feature of good object oriented design.
- Procedural programming will always have its place, but object oriented design is the way of the future.
- All modern programming languages feature the ability to code with objects and methods.
- A good understanding of object oriented design will greatly enhance a modern day programmers ability to create efficient, modular code that is capable of being utilized and contributed to by teams of developers.
- Documentation is vitally important in terms of the development and maintenance of professional code.

In this unit plan, the following 21st Century themes and skills are addressed.

Check all that apply. **21st Century Themes**	*Indicate whether these skills are **E**-Encouraged, **T-**Taught, or **A**-Assessed in this unit by marking **E, T, A** on the line before the appropriate skill.* **21st Century Skills**

	X	**Global Awareness**		**E,T,A**	**Creativity and Innovation**
		Environmental Literacy		**E,T,A**	**Critical Thinking and Problem Solving**
	X	**Health Literacy**		**E,T,A**	**Communication**
		Civic Literacy		**E,T,A**	**Collaboration**
	X	**Financial, Economic, Business, and Entrepreneurial Literacy**			

Student Learning Targets/Objectives (Students will know/Students will understand)

1. Analyze a given technological product, system, or environment to understand how the engineering design process and design specification limitations influenced the final solution.
1. Develop methods for creating possible solutions, modeling and testing solutions, and modifying proposed design in the solution of a technological problem using hands-on activities.
2. Practice the safe use of tools and equipment.
3. Working in cross-curricular areas to solve problems.
4. Given method and object specifications and documentation, create the appropriate code to solve a problem in a variety of curricular areas including math, science, engineering and english.
5. Research, documentation, and communication skills needed to produce a working software program.
6. Work collaboratively in a lab environment.
7. Identify and differentiate between a procedural programming approach and an object oriented one.
8. Describe how object oriented programs can be developed to solve problems.

Assessments (Pre, Formative, Summative, Other) ***Denote required common assessments with an ****

Throughout the course there is a strong emphasis on Object Oriented Programming and related concepts such as Data Encapsulation and Information Hiding. Students will work from a program specification to build their own creative solutions to problems. By its very nature the course is highly differentiated and each student is free to develop advanced features and enhancements beyond those required for a basic solution. The introduction of GUI (Graphic User Interfaces) early in the course further increases student options for creating exceptional graphics based applets and applications in addition or in place of a console based solution. Backwards Design in the discussion of a program solution will be featured regularly as we carefully consider which type of designs will provide for optimal solutions given the tools that students have acquired to date.

Formative and summative evaluations in the form of worksheets, quizzes and tests - both paper based and computer based will be utilized as needed to reinforce learning objectives and evaluate student progress.

Teaching and Learning Activities	
Activities	• Lecture and class discussion. • Video and multimedia presentations. • Build a console, applet and GUI based application. • Review and extend program functionality. • Group and collaborative work. • Student presentations of projects.
Differentiation Strategies	• Individual and collaborative research, design and problem solving • Student interest and skill level assessment • Individual, small group, and large group instruction • Media presentations and guest speakers • Student presentations and Flipped Lessons Differentiation Strategies for Special Education Students Differentiation Strategies for Gifted and Talented Students Differentiation Strategies for ELL Students Differentiation Strategies for At Risk Students
Resources	

- http://www.state.nj.us/education/cccs/
- http://www.corestandards.org/ELA-Literacy
- http://www.state.nj.us/education/cccs/

Wayne School District

Curriculum Format

Content Area/ Grade Level/ Course:	**Mathematics** **10 -12** **Honors Programming**
Unit Plan Title:	**Unit 4. GUI Applications**
Time Frame	**6 Weeks**

Anchor Standards/Domain*   *i.e: ELA: reading, writing i.e.: Math: Number and Operations in Base 10

Data and Analysis	DA
Algorithms and Programming	AP
Engineering Design	ED
21st Century Life Skills: Career Preparation	9.2.12.C
Career & Technical Education: Information Technology	9.3.IT-PRG

Unit Summary

Unit 4. GUI Applications and Applets

- What is the difference between an applet and application and what is a Graphical User Interface?
- Look at an existing GUI from an application and applet point of view.
- Analyze the functions of each component and how to use them.
- Attach listeners.
- Problem solve and add an atheistically looking GUI that will guide a user through using a program.
- Create a useful interface

Standard Number(s)

8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.
8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects
8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and Accessible
8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.
8.2.12.ED.2: Create scaled engineering drawings for a new product or system and make modification to increase optimization based on feedback.
9.2.12.CAP.6: Identify transferable skills in career choices and design alternative career plans based on those skills.
9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a)
9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).
WHST.11-12.4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
WHST.11-12.5. Develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach, focusing on addressing what is most significant for a specific purpose and audience.
WHST.11-12.6. Use technology, including the Internet, to produce, share, and update writing products in response to ongoing feedback, including new arguments or information.
NJSLSA.SL1. - Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.
SL.11-12.4 - Present information, findings and supporting evidence clearly, concisely, and logically. The content, organization, development, and style are appropriate to task, purpose, and audience.
HS-ETS1-2. - Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
HS-ETS1-4. - Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Essential Question(s)

- What is the difference between an applet and application and what is a Graphical User Interface?
- What is a component and how can you use them to create a GUI?
- What is a listener? Give examples of them all around us.
- How does a component effect they type of listener a programmer can use?
- How can a programmer use logic to make a listener perform more than one action?
- How does the component layout affect a user's experience?

Enduring Understandings

In this unit plan, the following 21st Century themes and skills are addressed.

Check all that apply. **21st Century Themes**		*Indicate whether these skills are **E**-Encouraged, **T**-Taught, or **A**-Assessed in this unit by marking **E, T, A** on the line before the appropriate skill.* **21st Century Skills**	
X	**Global Awareness**	**E,T,A**	**Creativity and Innovation**
	Environmental Literacy	**E,T,A**	**Critical Thinking and Problem Solving**
	Health Literacy	**E,T,A**	**Communication**
X	**Civic Literacy**	**E,T,A**	**Collaboration**
X	**Financial, Economic, Business, and Entrepreneurial Literacy**		

Student Learning Targets/Objectives (Students will know/Students will understand)

Assessments (Pre, Formative, Summative, Other) ***Denote required common assessments with an ****

Throughout the course there is a strong emphasis on Object Oriented Programming and related concepts such as Data Encapsulation and Information Hiding. Students will work from a program specification to build their own creative solutions to problems. By its very nature the course is highly differentiated and each student is free to develop advanced features and enhancements beyond those required for a basic solution. The introduction of GUI (Graphic User Interfaces) early in the course further increases student options for creating exceptional graphics based applets and applications in addition or in place of a console based solution. Backwards Design in the discussion of a program solution will be featured regularly as we carefully consider which type of designs will provide for optimal solutions given the tools that students have acquired to date.

Formative and summative evaluations in the form of worksheets, quizzes and tests - both paper based and computer based will be utilized as needed to reinforce learning objectives and evaluate student progress.

Teaching and Learning Activities	
Activities	• Lecture and class discussion. • Video and multimedia presentations. • Build a console, applet and GUI based application. • Review and extend program functionality. • Group and collaborative work. • Student presentations of projects.
Differentiation Strategies	• Individual and collaborative research, design and problem solving • Student interest and skill level assessment (Learning Style Assessment results) • Individual, small group, and large group instruction • Media presentations and guest speakers • Student presentations and Flipped Lessons Differentiation Strategies for Special Education Students Differentiation Strategies for Gifted and Talented Students Differentiation Strategies for ELL Students Differentiation Strategies for At Risk Students
Resources	

- http://www.state.nj.us/education/cccs/
- http://www.corestandards.org/ELA-Literacy
- http://www.state.nj.us/education/cccs/

Wayne School District
Curriculum Format

Content Area/ Grade Level/ Course:	**Mathematics / Computer Science 10-12 Honors Programming**
Unit Plan Title:	**Unit 5. Basic data structures including arrays**
Time Frame	**6 Weeks**

Anchor Standards/Domain*   *i.e: ELA: reading, writing i.e.: Math: Number and Operations in Base 10

Data and Analysis	DA
Algorithms and Programming	AP
Engineering Design	ED
21st Century Life Skills: Career Preparation	9.2.12.C
Career & Technical Education: Information Technology	9.3.IT-PRG

Unit Summary

Unit 5. Basic data structures including arrays

A. Strings as an array of chars
   1. Several String based programs involving String traversal and looping to reinforce key concepts.

B. Array declaration, instantiation and traversal

C. A variety of array based programs will be discussed, tested and built to specification.

Standard Number(s)

8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
8.1.12.DA.3: Translate between decimal numbers and binary numbers.
8.1.12.DA.4: Explain the relationship between binary numbers and the storage and use of data in a computing device.
8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.
8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects
8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and
Accessible
8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.
8.2.12.ED.2: Create scaled engineering drawings for a new product or system and make modification to increase optimization based on feedback.
9.2.12.CAP.6: Identify transferable skills in career choices and design alternative career plans based on those skills.
9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a)
9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).
WHST.11-12.4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
WHST.11-12.5. Develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach, focusing on addressing what is most significant for a specific purpose and audience.
WHST.11-12.6. Use technology, including the Internet, to produce, share, and update writing products in response to ongoing feedback, including new arguments or information.
NJSLSA.SL1. - Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.
SL.11-12.4 - Present information, findings and supporting evidence clearly, concisely, and logically. The content, organization, development, and style are appropriate to task, purpose, and audience.

HS-ETS1-2. - Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
HS-ETS1-4. - Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Essential Question(s)

- How is the String data type similar to an Array of chars?
- What does it mean to declare, instantiate and initialize an Array?
- Why is it so important to learn about Arrays?

Enduring Understandings

- Strings, Arrays and For Loop traversal are all interconnected concepts.
- Arrays allow us to keep track of several pieces of data in one organized, orderly structure.
- Arrays are a powerful data structure that allow us to create more flexible and robust software.

In this unit plan, the following 21st Century themes and skills are addressed.

Check all that apply. **21st Century Themes**		*Indicate whether these skills are **E**-Encouraged, **T-**Taught, or **A**-Assessed in this unit by marking **E, T, A** on the line before the appropriate skill.* **21st Century Skills**	
X	**Global Awareness**	**E,T,A**	**Creativity and Innovation**
	Environmental Literacy	**E,T,A**	**Critical Thinking and Problem Solving**
	Health Literacy	**E,T,A**	**Communication**
X	**Civic Literacy**	**E,T,A**	**Collaboration**
X	**Financial, Economic, Business, and Entrepreneurial Literacy**		

Student Learning Targets/Objectives (Students will know/Students will understand)

1. Become familiar with the demands and qualifications for a career in computer science.
2. Participate in a series of in-class activities and projects that emphasize and build upon the skills utilized in computer programming.
3. Further refine skills using a software development environment and key programming constructs such as objects, random numbers and arrays.
4. Create programs using a program specification.
5. Completed programs will be saved to an electronic Student Portfolio.
6. Develop the ability to design and implement algorithms to solve problems by using arrays.
7. Create programs that model real world games and situations, as well as mathematical calculations.
8. Develop methods for creating possible solutions, modeling and testing solutions, and modifying proposed design in the solution of a technological problem using hands-on activities.
9. Create pictures to help visualize an array of data and the actions that need to be taken.
10. Work cooperatively in a lab environment
11. Design, build, test and modify a program.

Assessments (Pre, Formative, Summative, Other) ***Denote required common assessments with an ****

Throughout the course there is a strong emphasis on Object Oriented Programming and related concepts such as Data Encapsulation and Information Hiding.
Students will work from a program specification to build their own creative solutions to problems. By its very nature the course is highly differentiated and each student is free to develop advanced features and enhancements beyond those required for a basic solution. The introduction of GUI (Graphic User Interfaces) early in the course further increases student options for creating exceptional graphics based applets and applications in addition or in place of a console based solution. Backwards Design in the discussion of a program solution will be featured regularly as we carefully consider which type of designs will provide for optimal solutions given the tools that students have acquired to date.

Formative and summative evaluations in the form of worksheets, quizzes and tests - both paper based and computer based will be utilized as needed to reinforce learning objectives and evaluate student progress.

Teaching and Learning Activities

Activities	• Lecture/discussion • Create a picture of an array to help determine necessary code. • Hands-on programming activities • Choose the best approach, tools and data type for a given program specification • Testing and presentation of completed programs.
Differentiation Strategies	• Individual and collaborative research, design and problem solving • Student interest and skill level assessment (Learning Style Assessment results) • Individual, small group, and large group instruction • Media presentations and guest speakers • Student presentations and Flipped Lessons • Students always have the opportunity to make their programs better and extend program functionality. Differentiation Strategies for Special Education Students Differentiation Strategies for Gifted and Talented Students Differentiation Strategies for ELL Students Differentiation Strategies for At Risk Students

Resources

- **http://www.state.nj.us/education/cccs/**
- **http://www.corestandards.org/ELA-Literacy**
- **http://www.state.nj.us/education/cccs/**

Wayne School District
Curriculum Format

Content Area/ Grade Level/ Course:	**Mathematics** **10 -12** **Honors Programming**
Unit Plan Title:	**Unit 6. Game Logic and Design**
Time Frame	**6 Weeks**

Anchor Standards/Domain*   *i.e: ELA: reading, writing i.e.: Math: Number and Operations in Base 10

Data and Analysis	DA
Algorithms and Programming	AP
Engineering Design	ED
Computer Systems	CS
21st Century Life Skills: Career Preparation	9.2.12.C
Career & Technical Education: Information Technology	9.3.IT-PRG

Unit Summary

Unit 6. Game Logic and Design

- Introduction to drawing shapes using awt
- Combining awt and swing
- Drawing an image to the screen.
- Basic collision detection with stationary objects and vector movement.
- Using timers
- Movement of objects using the timer and keyboard.
- Introduction to sprites and animation.
- The game loop and game logic

Standard Number(s)

8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
8.1.12.DA.5: Create data visualizations from large data sets to summarize, communicate, and support different interpretations of real-world phenomena.
8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.
8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.4: Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects
8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.1.12.AP.7: Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users.
8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and Accessible
8.1.12.AP.9: Collaboratively document and present design decisions in the development of complex programs.
8.2.12.ED.2: Create scaled engineering drawings for a new product or system and make modification to increase optimization based on feedback.
9.2.12.CAP.6: Identify transferable skills in career choices and design alternative career plans based on those skills.
9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
9.4.12.CI.3: Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a)
9.4.12.CT.4: Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
9.4.12.TL.4: Collaborate in online learning communities or social networks or virtual worlds to analyze and propose a resolution to a real-world problem (e.g., 7.1.AL.IPERS.6).
WHST.11-12.4. Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
WHST.11-12.5. Develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach, focusing on addressing what is most significant for a specific purpose and audience.
WHST.11-12.6. Use technology, including the Internet, to produce, share, and update writing products in response to ongoing feedback, including new arguments or information.
NJSLSA.SL1. - Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.
SL.11-12.4 - Present information, findings and supporting evidence clearly, concisely, and logically. The content, organization, development, and style are appropriate to task, purpose, and audience.
HS-ETS1-2. - Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
HS-ETS1-4. - Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

Essential Question(s)

- How is using paint and paintComponents different?
- What is the content pane and how do you access it?
- What is the graphics object and how do you repaint it?
- How are some ways a programmer can move an image around the screen?
- What is collision detection and how do you make a bonding box using coordinates?
- What is a game loop? And how can the programmer use a timer to create one?

Enduring Understandings

- paintComponents is used to draw objects over existing swing components.
- The content pane is the layer drawings should be placed on when dealing with the graphic object.
- Using vectors, the programmer can change the position objects on the screen. This gives the illusion of movement.
- Collision detection is the idea of one object taking up part of, or the entire, space of another object. Using inequalities, a programmer can create a bounding box and compare the coordinates.
- The game loop drives the game forward. It's an "infinite" cycle that starts, modifies, and draws the game, until it is closed.

In this unit plan, the following 21st Century themes and skills are addressed.

	Check all that apply. **21st Century Themes**		*Indicate whether these skills are **E**-Encouraged, **T**-Taught, or **A**-Assessed in this unit by marking **E, T, A** on the line before the appropriate skill.* **21st Century Skills**
X	**Global Awareness**	**E,T,A**	**Creativity and Innovation**
	Environmental Literacy	**E,T,A**	**Critical Thinking and Problem Solving**
	Health Literacy	**E,T,A**	**Communication**
X	**Civic Literacy**	**E,T,A**	**Collaboration**
X	**Financial, Economic, Business, and Entrepreneurial Literacy**		

Student Learning Targets/Objectives (Students will know/Students will understand)

- Develop methods for creating possible solutions, modeling and testing solutions, and modifying proposed design in the solution of a technological problem using hands-on activities.
- Use swing and awt to draw to the screen.
- Draw and move and image using vectors.
- Use timers and the keyboard to move objects.
- Create collision boxes around an objects.
- Create a game loop and understand the paint repaint process.
- Have an understanding of what makes a quality game.

Assessments (Pre, Formative, Summative, Other) ***Denote required common assessments with an ****

Throughout the course there is a strong emphasis on Object Oriented Programming and related concepts such as Data Encapsulation and Information Hiding. Students will work from a program specification to build their own creative solutions to problems. By its very nature the course is highly differentiated and each student is free to develop advanced features and enhancements beyond those required for a basic solution. The introduction of GUI (Graphic User Interfaces) early in the course further increases student options for creating exceptional graphics based applets and applications in addition or in place of a console based solution. Backwards Design in the discussion of a program solution will be featured regularly as we carefully consider which type of designs will provide for optimal solutions given the tools that students have acquired to date.

Formative and summative evaluations in the form of worksheets, quizzes and tests - both paper based and computer based will be utilized as needed to reinforce learning objectives and evaluate student progress.

Teaching and Learning Activities

Activities	• Lecture and class discussion. • Video and multimedia presentations. • Build a console, applet and GUI based application. • Review and extend program functionality. • Group and collaborative work. • Student presentations of projects.
Differentiation Strategies	• Individual and collaborative research, design and problem solving • Student interest and skill level assessment (Learning Style Assessment results) • Individual, small group, and large group instruction • Media presentations and guest speakers • Student presentations and Flipped Lessons Differentiation Strategies for Special Education Students Differentiation Strategies for Gifted and Talented Students Differentiation Strategies for ELL Students Differentiation Strategies for At Risk Students

Resources

- http://www.state.nj.us/education/cccs/
- http://www.corestandards.org/ELA-Literacy
- http://www.state.nj.us/education/cccs/