

PUBLIC SCHOOLS OF EDISON TOWNSHIP
DIVISION OF CURRICULUM AND INSTRUCTION

**COMPUTER SCIENCE:
ADVANCED PLACEMENT**

Length of Course: Term

Elective/Required: Elective

School: High Schools

Student Eligibility: Grades 11-12

Credit Value: 5 Credits

Date Approved: 11/22/10

THE PUBLIC SCHOOLS OF EDISON TOWNSHIP

MISSION STATEMENT

The Public Schools of Edison Township ensure that all students achieve at the highest level of academic success through the New Jersey Core Curriculum Content Standards and in partnership with the community, through a safe, supportive learning environment. This promotes self-worth and encourages productive contributions to a diverse, technological and constantly evolving global society. The district will maintain a staff of professional educators who support the New Jersey Core Curriculum Content Standards and the New Jersey Standards for Professional Development.

TABLE OF CONTENTS

Statement of Purpose	3
Introduction	4
Course Objectives	6
Suggested Time Schedule	7

Course Outline

Unit I: Object-Oriented Program Design	8
Unit II: Program Implementation and Analysis	14
Unit III: Standard Data Structures	21
Unit IV: Standard Algorithms	29
Essential Instructional Behavior (Draft 14)	30

Modifications will be made to accommodate IEP mandates for classified students.

STATEMENT OF PURPOSE

This course is designed to prepare students for the Computer Science A advanced placement examination.

Students will use the Java programming language and eclipse IDE. The prerequisite for this course is Computer Science Java.

This curriculum guide was revised and updated by:

Florene Quan - John P. Stevens High School

Coordinated by:

Jessica Lewis - Supervisor of Mathematics, Edison High School

Vincent Ciraulo - Supervisor of Mathematics, John P. Stevens High School

Introduction

The most precious resource teachers have is time. Regardless of how much time a course is scheduled for, it is never enough to accomplish all that one would like. Therefore, it is imperative that teachers utilize the time they have wisely in order to maximize the potential for all students to achieve the desired learning.

High quality educational programs are characterized by clearly stated goals for student learning, teachers who are well-informed and skilled in enabling students to reach those goals, program designs that allow for continuous growth over the span of years of instruction, and ways of measuring whether students are achieving program goals.

The Edison Township School District Curriculum Template

The Edison Township School District has embraced the backward-design model as the foundation for all curriculum development for the educational program. When reviewing curriculum documents and the Edison Township curriculum template, aspects of the backward-design model will be found in the stated enduring *understandings/essential questions*, *unit assessments*, and *instructional activities*. Familiarization with backward-design is critical to working effectively with Edison's curriculum guides.

Guiding Principles: What is Backward Design? What is Understanding by Design?

"Backward design" is an increasingly common approach to planning curriculum and instruction. As its name implies, "backward design" is based on defining clear goals, providing acceptable evidence of having achieved those goals, and then working 'backward' to identify what actions need to be taken that will ensure that the gap between the current status and the desired status is closed.

Building on the concept of backward design, Grant Wiggins and Jay McTighe (2005) have developed a structured approach to planning programs, curriculum, and instructional units. Their model asks educators to state goals; identify deep understandings, pose essential questions, and specify clear evidence that goals, understandings, and core learning have been achieved.

Program based on backward design use desired results to drive decisions. With this design, there are questions to consider, such as: What should students understand, know, and be able to do? What does it look like to meet those goals? What kind of program will result in the outcomes stated? How will we know students have achieved that result? What other kinds of evidence will tell us that we have a quality program? These questions apply regardless of whether they are goals in program planning or classroom instruction.

The backward design process involves three interrelated stages for developing an entire curriculum or a single unit of instruction. The relationship from planning to curriculum design, development, and implementation hinges upon the integration of the following three stages.

Stage I: Identifying Desired Results: Enduring understandings, essential questions, knowledge and skills need to be woven into curriculum publications, documents, standards, and scope and sequence materials. Enduring understandings identify the “big ideas” that students will grapple with during the course of the unit. Essential questions provide a unifying focus for the unit and students should be able to answer more deeply and fully these questions as they proceed through the unit. Knowledge and skills are the “*stuff*” upon which the understandings are built.

Stage II: Determining Acceptable Evidence: Varied types of evidence are specified to ensure that students demonstrate attainment of desired results. While discrete knowledge assessments (e.g.: multiple choice, fill-in-the-blank, short answer, etc...) will be utilized during an instructional unit, the overall unit assessment is performance-based and asks students to demonstrate that they have mastered the desired understandings. These culminating (summative) assessments are authentic tasks that students would likely encounter in the real-world after they leave school. They allow students to demonstrate all that they have learned and can do. To demonstrate their understandings students can explain, interpret, apply, provide critical and insightful points of view, show empathy and/or evidence self-knowledge. Models of student performance and clearly defined criteria (i.e.: rubrics) are provided to all students in advance of starting work on the unit task.

Stage III: Designing Learning Activities: Instructional tasks, activities, and experiences are aligned with stages one and two so that the desired results are obtained based on the identified evidence or assessment tasks. Instructional activities and strategies are considered only once stages one and two have been clearly explicated. Therefore, congruence among all three stages can be ensured and teachers can make wise instructional choices.

At the curricular level, these three stages are best realized as a fusion of research, best practices, shared and sustained inquiry, consensus building, and initiative that involves all stakeholders. In this design, administrators are instructional leaders who enable the alignment between the curriculum and other key initiatives in their district or schools. These leaders demonstrate a clear purpose and direction for the curriculum within their school or district by providing support for implementation, opportunities for revision through sustained and consistent professional development, initiating action research activities, and collecting and evaluating materials to ensure alignment with the desired results. Intrinsic to the success of curriculum is to show how it aligns with the overarching goals of the district, how the document relates to district, state, or national standards, what a high quality educational program looks like, and what excellent teaching and learning looks like. Within education, success of the educational program is realized through this blend of commitment and organizational direction.

Course Objectives

Students will be able to:

- Understand the main principles of object-oriented software design and programming.
- Write code fluently in Java in a well-structured fashion and in good style.
- Use Java library packages and classes within the scope of the AP Java Subset.
- Understand the concepts of an algorithm; implement algorithms in Java using conditional and iterative control statements and recursion.
- Use common sorting and searching algorithms.
- Determine time and space efficiencies of algorithms using Big O notation.
- Understand standard data structures including one-dimensional and two-dimensional arrays, lists, stacks, queues, binary search trees, hash tables and priority queues and to use the classes and interfaces for linked lists, stacks, queues and binary trees provided by the College Board.
- Use the GridWorld case study or current case study and the accompanying exercises and questions provided by the College Board.

SUGGESTED TIME LINE

	<u>UNIT</u>	<u># PERIODS</u>
Unit I:	Object-Oriented Program Design	30
Unit II:	Program Implementation and Analysis	30
Unit III:	Standard Data Structures	100
Unit IV:	Standard Algorithms	20
	Total Class Periods	180

Object-Oriented Program Design

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Student will be able to learn a brief history of computers, how hardware and software make up computer architecture, understand the binary representation of data, discuss the evolution of programming languages, describe the software development process and discuss the fundamental concepts of object-oriented programming.

Essential Questions: How can a computer be used to solve real-world problems?

Unit Assessment: Programming assignments and chapter test

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
<p>Analyze a given technological product, system, or environment to understand how the engineering design process and design specification limitations influenced the final solution. 8.1.12.B.1</p> <ul style="list-style-type: none"> Recognize components of computer architecture such as hardware and software components, the Central Processing Unit, Internal Memory, Auxiliary Storage devices. Understand Binary Representation of Information and Computer Memory such as binary, octal and hexadecimal, integer and its limitations, floating point and its imprecision, boolean, character and string, and ASCII vs. Unicode. 	<ul style="list-style-type: none"> A computer is comprised of hardware and software components. Memory in a computer is translated and stored in a binary, octal and hexadecimal. Programming languages have evolved from Machine Language, Assembly Language to High-level languages. The software development process is in six steps. There are legal, ethical and privacy issues with a computer. 	<ul style="list-style-type: none"> Identify the components of a computer Translate the decimal representation of a number to its corresponding binary, octal and hexadecimal equivalent. Download eclipse from www.eclipse.org. Create a java project in eclipse. Write classes in eclipse. 	<ul style="list-style-type: none"> Use PowerPoint slides provided with Teacher Resources Select several programming projects from the end of the chapter Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program 	<ul style="list-style-type: none"> Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. Continually emphasize grading exam standards developed by College Board.

Object-Oriented Program Design (Cont.)

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
<ul style="list-style-type: none"> Learn the history of programming languages such as Machine Language, Assembly Languages and other High Level Languages. Utilize proper software development process of understanding user request, analyzing, designing, writing implementation code, integrating software product to user's environment and maintenance of software product. Use a different development environment that can edit, compile and execute. Use text editors and JDK. Use an Integrated Development Environment such as eclipse. Discuss writing programs using a procedural Approach vs. an Object-Oriented Approach. Responsible use of computer systems. Discuss privacy issues, ethical and legal use of a computer, social implications of computer use and system reliability. 				

Object-Oriented Program Design (Cont.)**Resources:** Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and related Teacher Resources.
- Number Systems - <http://www.purplemath.com/modules/numbase.htm>
- Computer Fundamentals - <http://www.adita.com/one-a.htm>
- Integrated Development Environment – <http://www.eclipse.org>
- www.java.sun.com

Instructional Adjustments: Modifications, student difficulties, possible misunderstandings

- Circulate around classroom during Lab and offer one-on-one assistance for struggling students

Object-Oriented Program Design

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Students will be able to discuss concepts of Object-Oriented solutions to problems.

Essential Questions: What are possible designs of solutions for real-world problems?

Unit Assessment: Programming assignments and chapter test

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
<p>Create and manipulate information, independently and/or collaboratively, to solve problems and design and develop products. 8.1.12 B 9</p> <ul style="list-style-type: none"> Identify concepts related to program design which include: the purpose and goals of the problem, the sub-tasks to be performed, the ADT's and operations needed to solve the problem, the reusable components from existing code, decomposing subprograms, choosing proper data structures and algorithms, and designing the user interface. 	<ul style="list-style-type: none"> The goal of designing a piece of software is to correctly solve a problem that is understandable, can be adapted to changing circumstances and has the potential to be reused in whole or in part. Encapsulation is the combination of resources and behaviors into a single entity. Classes consists of instance variables and methods Design of classes that follow the principle of information hiding. 	<ul style="list-style-type: none"> Break down a problem into smaller, more manageable parts. Determine an object's state and behavior. Discuss possible ADT's appropriate for solving the problem. Develop algorithms for solving the problem. Execute programs that test the boundaries of the Java class. 	<ul style="list-style-type: none"> Use PowerPoint slides provided with Teacher Resources Select several programming projects from the end of the chapter Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program 	<ul style="list-style-type: none"> Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. Continually emphasize grading exam standards developed by College Board.

Object-Oriented Program Design (Cont.)

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
<ul style="list-style-type: none"> • Understand the purpose and goals of the problem, the sub-tasks to be performed, the ADT's and operations needed to solve the problem, the reusable components from existing code, decomposing subprograms, choosing proper data structures and algorithms, designing the user interface. • Perform tasks related to program design. Specifying the purpose and goals of the problem. Identifying and isolating the sub-tasks to be performed. Determining the ADT's and operations needed to solve the problem. Identifying and using the reusable components from existing code. Design the user interface. Apply data abstraction and encapsulation. • Testing modules in isolation. Identifying boundary cases and generating appropriate test data. • Identify techniques of debugging programs which include categorizing errors as syntax, run-time or logic, identifying and correcting errors and using debugging techniques. 				

Object-Oriented Program Design (Cont.)

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
<ul style="list-style-type: none"> Analyzing algorithms by informal comparisons of running times and counting exact calculation of statement execution. 				
Resources: Essential Materials, Supplementary Materials, Links to Best Practices <ul style="list-style-type: none"> Textbook and related Teacher Resources. Computer Fundamentals - http://www.adita.com/one-a.htm Integrated Development Environment – http://www.eclipse.org www.java.sun.com GridWorld Case Study - www.apcentral.collegeboard.com 			Instructional Adjustments: Modifications, student difficulties, possible misunderstandings <ul style="list-style-type: none"> Circulate around classroom during Lab and offer one-on-one assistance for struggling students 	

Program Implementation and Analysis

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Students will be able to define Java classes.

Essential Questions: How can real-world objects be represented as Java classes?

Unit Assessment: Programming assignments and chapter test

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
<p>Create and manipulate information, independently and/or collaboratively, to solve problems and design and develop products. 8.1.12.B.9</p> <ul style="list-style-type: none"> Understanding characteristics of classes such as encapsulation and information hiding. Understanding concepts of a Java class such as state and behavior; how one class can make many objects; and the ability to instantiate an object. Recognize different types of data; using appropriate data types; apply proper protection of data by using private, protected and public modifiers. Understand the difference between primitive data type vs. objects. 	<ul style="list-style-type: none"> The anatomy of a class; public and private instance variable, and public and private methods. The use of inheritance to create “is-a” relationship and “has-a” relationship Knowing when it is appropriate to use primitive data types or a class. 	<ul style="list-style-type: none"> Using eclipse to create a Java class. Reading examples of pre-written classes and modifying them. To be able to create objects from classes and call appropriate methods from the classes. To review GridWorld as an example of a complete application of classes 	<ul style="list-style-type: none"> Use PowerPoint slides provided with Teacher Resources Select several programming projects from the end of the chapter Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program 	<ul style="list-style-type: none"> Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. Continually emphasize grading exam standards developed by College Board.

Program Implementation and Analysis (Cont.)

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
<ul style="list-style-type: none"> To be able to translate behavior of an object into Java methods; to write and invoke methods and recognize “setter” vs. “getter” methods. To be able to call appropriate algorithms based on a class hierarchy. Designing, structuring and implementing of related and interacting classes. Understanding the scope and lifetime of variables, parameters and objects. Using inheritance to create superclasses and subclasses in order to solve problems. 				
<p>Resources: Essential Materials, Supplementary Materials, Links to Best Practices</p> <ul style="list-style-type: none"> Textbook and related Teacher Resources. Computer Fundamentals - http://www.adita.com/one-a.htm Integrated Development Environment – http://www.eclipse.org www.java.sun.com GridWorld Case Study - www.apcentral.collegeboard.com 			<p>Instructional Adjustments: Modifications, student difficulties, possible misunderstandings</p> <ul style="list-style-type: none"> Circulate around classroom during Lab and offer one-on-one assistance for struggling students 	

Program Implementation and Analysis (Cont.)

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Student will be able to understand Java as a programming language, to understand benefits of Java, to understand concept of Object Oriented Programming.

Essential Questions: How are Java classes most effectively designed to take advantage of reusing existing classes?

Unit Assessment: Programming assignments and chapter test

		Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points	
<p>Create and manipulate information, independently and/or collaboratively, to solve problems and design and develop products. 8.1.12.B.9</p> <ul style="list-style-type: none"> • Understanding Java as a programming language including the role of Java Virtual Machine and Java byte code. • Understand the benefits of Java and its features such as security, robustness, portability and its use on the Internet. • Understand the concepts of Object Oriented Programming including encapsulation, inheritance, information-hiding and ability to reuse code. 	<ul style="list-style-type: none"> • To build on the concept of inheritance and the “is-a” relationship and the “has-a” relationship 	<ul style="list-style-type: none"> • Use the extends keyword. • To write and design simple Java classes. • To be able to sketch a hierarchy of classes. 	<ul style="list-style-type: none"> • Use PowerPoint slides provided with Teacher Resources • Select several programming projects from the end of the chapter • Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program 	<ul style="list-style-type: none"> • Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. • Continually emphasize grading exam standards developed by College Board. 	
<p>Resources: Essential Materials, Supplementary Materials, Links to Best Practices</p> <ul style="list-style-type: none"> • Textbook and related Teacher Resources. • Computer Fundamentals - http://www.adita.com/one-a.htm • Integrated Development Environment – http://www.eclipse.org • www.java.sun.com • GridWorld Case Study - www.apcentral.collegeboard.com 			<p>Instructional Adjustments: Modifications, student difficulties, possible misunderstandings</p> <ul style="list-style-type: none"> • Circulate around classroom during Lab and offer one-on-one assistance for struggling students 		

Program Implementation and Analysis (Cont.)

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Students will be able to make appropriate use of static variables and methods, to understand the role of Java interfaces, and to understand inheritance exceptions.

Essential Questions: What behaviors are common among various Java classes?

Unit Assessment: Programming assignments and chapter test

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
<p>Create and manipulate information, independently and/or collaboratively, to solve problems and design and develop products. 8.1.12.B.9</p> <ul style="list-style-type: none"> • Understanding how to categorize various certain variables and methods so that they are independent of each other. • Understanding the need to create generalized classes. • Understand the concept of inheritance as it applies to Java classes. • Understanding the source of standard exceptions. 	<ul style="list-style-type: none"> • Some variables and or methods remain static regardless of how a class is used • Some classes and or interfaces act as a shell that will be implemented at a future time • Inheritance allows reuse of code • Java provides a class to handle programming errors 	<ul style="list-style-type: none"> • Use the static keyword on variables and methods • Implement interfaces and abstract classes • Use the extends keyword to create a parent class (super class) and a child class (sub class). • Determine a solution based on a Java Exception 	<ul style="list-style-type: none"> • Use PowerPoint slides provided with Teacher Resources • Select several programming projects from the end of the chapter • Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program 	<ul style="list-style-type: none"> • Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. • Continually emphasize grading exam standards developed by College Board.

Program Implementation and Analysis (Cont.)**Resources:** Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and related Teacher Resources.
- Computer Fundamentals - <http://www.adita.com/one-a.htm>
- Integrated Development Environment – <http://www.eclipse.org>
- www.java.sun.com
- GridWorld Case Study - www.apcentral.collegeboard.com

Instructional Adjustments: Modifications, student difficulties, possible misunderstandings

- Circulate around classroom during Lab and offer one-on-one assistance for struggling students

Program Implementation and Analysis (Cont.)

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Student will be able to use a variety of Java classes, interfaces and methods.

Essential Questions When is it useful to use a wrapper class and the Comparable interface?

Unit Assessment: Programming assignments and chapter test

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
<p>Create and manipulate information, independently and/or collaboratively, to solve problems and design and develop products. 8.1.12.B.9</p> <ul style="list-style-type: none"> Ability to choose correct methods from a variety of objects in order to solve problems. 	<ul style="list-style-type: none"> java.lang.Object is the highest level class and contains these methods: <ol style="list-style-type: none"> equals (Object other) toString () hashCode () java.lang.Comparable interface contains the compareTo method java.lang.Integer is a wrapper class that contains these methods: <ol style="list-style-type: none"> Integer (int value) intValue () equals (object other) toString () compareTo (object other) java.lang.Double is a wrapper class that contains these methods: <ol style="list-style-type: none"> Double doubleValue () equals (object other) toString () compareTo (object other) 	<ul style="list-style-type: none"> To make appropriate use of wrapper classes. To recognize Object as the top level class in the Java hierarchy To recognize when to use .compareTo and when to use .equals To understand that String is a class 	<ul style="list-style-type: none"> Use PowerPoint slides provided with Teacher Resources Select several programming projects from the end of the chapter Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program 	<ul style="list-style-type: none"> Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. Continually emphasize grading exam standards developed by College Board.

Program Implementation and Analysis (Cont.)

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
	<ul style="list-style-type: none"> • class java.lang.String contains these methods: <ul style="list-style-type: none"> a. compareTo (Object other) b. equals (Object other) c. length () d. substring (from, to), substring e. indexOf (s) • class java.lang.Math contains these static methods <ul style="list-style-type: none"> a. abs (x) b. pow (base, exponent) c. sqrt (x) • class java.util.Random <ul style="list-style-type: none"> a. nextInt (n) b. nextDouble () 			
<p>Resources: Essential Materials, Supplementary Materials, Links to Best Practices</p> <ul style="list-style-type: none"> • Textbook and related Teacher Resources. • Computer Fundamentals - http://www.adita.com/one-a.htm • Integrated Development Environment – http://www.eclipse.org • www.java.sun.com • GridWorld Case Study - www.apcentral.collegeboard.com 			<p>Instructional Adjustments: Modifications, student difficulties, possible misunderstandings</p> <ul style="list-style-type: none"> • Circulate around classroom during Lab and offer one-on-one assistance for struggling students 	

Standard Data Structures

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Student will be able to declare, manipulate and use arrays of primitive data.

Essential Questions: How are large amounts of data handled in Java?

Unit Assessment: Programming assignments and chapter test

		Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points	
<p>Create and manipulate information, independently and/or collaboratively, to solve problems and design and develop products. 8.1.12.B.9</p> <ul style="list-style-type: none"> Ability to store and manage large amounts of data using arrays. 	<ul style="list-style-type: none"> Declare and manipulate simple arrays, traversing through arrays, adding and removing elements. Understand the difference between physical size and logical size. Declare and manipulate two-dimensional arrays an array references to arrays To pass arrays as parameters and invoke methods. Declare and manipulate an array of objects. 	<ul style="list-style-type: none"> To understand the difference between the length constant of an array and the length() method of a String. To avoid an Off-By-One-Error when working with loops and arrays. For example for(i=0;i<arr.length;i++) is correct, whereas for(i=0;i<=arr.length;i++) will generate error. To use the enhanced for loop, also called the for-each loop To visualize a two-dimensional array with a grid 	<ul style="list-style-type: none"> Use PowerPoint slides provided with Teacher Resources Select several programming projects from the end of the chapter Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program "Ragged" Arrays (new int [3] []) are not required. Neither is the concept of "array of arrays". However, students should know a[o].length is the number of columns in a two-dimensional array. 	<ul style="list-style-type: none"> Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. Continually emphasize grading exam standards developed by College Board. 	

Standard Data Structures (Cont.)

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
			<ul style="list-style-type: none"> Students are expected to understand exceptions Array Index Out of Bounds Exception and No Such Element Exception <p>Continually use GridWorld or current case study as a source of examples.</p>	
<p>Resources: Essential Materials, Supplementary Materials, Links to Best Practices</p> <ul style="list-style-type: none"> Textbook and related Teacher Resources. Computer Fundamentals - http://www.adita.com/one-a.htm Integrated Development Environment – http://www.eclipse.org www.java.sun.com GridWorld Case Study - www.apcentral.collegeboard.com 			<p>Instructional Adjustments: Modifications, student difficulties, possible misunderstandings</p> <ul style="list-style-type: none"> Circulate around classroom during Lab and offer one-on-one assistance for struggling students 	

Standard Data Structures(Cont.)

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Student will be able to declare, manipulate and use the ArrayList class to manage an array of objects and to use features of the String class.

Essential Questions: How are arrays of objects and their methods handled in an ArrayList? How is text handled in Java?

Unit Assessment: Programming assignments and chapter test

		Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points	
<p>Create and manipulate information, independently and/or collaboratively, to solve problems and design and develop products. 8.1.12.B.9</p> <ul style="list-style-type: none"> • Ability to store and manage large amounts of data using arrays. • Ability to search and sort data stored in arrays. • Ability to use and manipulate text in Java. 	<ul style="list-style-type: none"> • Use java.util.ArrayList to manage an array of objects. • Use java.util.List get, set, add, remove, isEmpty, size and indexOf methods. • Use String methods such as length, substring and indexOf. • Implement searching algorithms such as linear search and binary search • Implement sorting algorithm such as: selection sort, insertion sort, quick sort. 	<ul style="list-style-type: none"> • The compareTo method of the Comparable Interface comparable should be used with string objects, with searching algorithms and with wrapper classes. • Use of GridWorld or current case study as a source of examples. • Continually use sample AP multiple choice and free response questions throughout the year. 	<ul style="list-style-type: none"> • Use PowerPoint slides provided with Teacher Resources • Select several programming projects from the end of the chapter • Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program 	<ul style="list-style-type: none"> • Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. • Continually emphasize grading exam standards developed by College Board. 	
<p>Resources: Essential Materials, Supplementary Materials, Links to Best Practices</p> <ul style="list-style-type: none"> • Textbook and related Teacher Resources. • Computer Fundamentals - http://www.adita.com/one-a.htm • Integrated Development Environment – http://www.eclipse.org • www.java.sun.com • www.java.sun.com/docs/books/tutorial/java/index.html • GridWorld Case Study - www.apcentral.collegeboard.com 			<p>Instructional Adjustments: Modifications, student difficulties, possible misunderstandings</p> <ul style="list-style-type: none"> • Circulate around classroom during Lab and offer one-on-one assistance for struggling students 		

Standard Data Structures(Cont.)

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Student will be able to use basic features of a linked list data structure.

Essential Questions: How are features of a linked list used to manage large amounts of data?

Unit Assessment: Programming assignments and chapter test

		Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points	
<p>Create and manipulate information, independently and/or collaboratively, to solve problems and design and develop products. 8.1.12.B.9</p> <ul style="list-style-type: none"> Ability to store and manage large amounts of data using linked lists. 	<ul style="list-style-type: none"> Implement and use linked lists Definition (singly, doubly, circular) Use methods of list interface including add, remove and set. Use methods of ListIterator interface including Iterator, ListIterator, addFirst, addLast, getFirst, getLast, removeFirst, removeLast, hasNext, next and remove 	<ul style="list-style-type: none"> Traverse through a singly or doubly or circular linked list to locate, add or remove a node To use the ListNode class and a Linked List class in an application To use GridWorld as a source of example Continually use sample AP multiple choice and free response questions throughout the year. 	<ul style="list-style-type: none"> Use PowerPoint slides provided with Teacher Resources Select several programming projects from the end of the chapter Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program 	<ul style="list-style-type: none"> Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. Continually emphasize grading exam standards developed by College Board. 	
<p>Resources: Essential Materials, Supplementary Materials, Links to Best Practices</p> <ul style="list-style-type: none"> Textbook and related Teacher Resources. Computer Fundamentals - http://www.adita.com/one-a.htm Integrated Development Environment – http://www.eclipse.org www.java.sun.com www.java.sun.com/docs/books/tutorial/java/index.html GridWorld Case Study - www.apcentral.collegeboard.com 			<p>Instructional Adjustments: Modifications, student difficulties, possible misunderstandings</p> <ul style="list-style-type: none"> Circulate around classroom during Lab and offer one-on-one assistance for struggling students 		

Standard Data Structures(Cont.)

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Student will be able to understand and use features of Sets and Maps.

Essential Questions: How are features of Sets and Maps used to manage large amounts of data?

Unit Assessment: Programming assignments and chapter test

		Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points	
<p>Create and manipulate information, independently and/or collaboratively, to solve problems and design and develop products. 8.1.12.B.9</p> <ul style="list-style-type: none"> Ability to store and manage large amounts of data using sets and maps. 	<ul style="list-style-type: none"> Implement and use sets and methods of the Set interface including add, contains, remove, size and iterator Implement and use maps and methods of the Map interface including put, get, containsKey, size and keySet 	<ul style="list-style-type: none"> To understand the limitations of a Set (cannot have duplicates) To understand the limitations of a Map (must have a unique associated with every piece of data) 	<ul style="list-style-type: none"> Use PowerPoint slides provided with Teacher Resources Select several programming projects from the end of the chapter Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program. 	<ul style="list-style-type: none"> Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. Continually emphasize grading exam standards developed by College Board. 	
<p>Resources: Essential Materials, Supplementary Materials, Links to Best Practices</p> <ul style="list-style-type: none"> Textbook and related Teacher Resources. Computer Fundamentals - http://www.adita.com/one-a.htm Integrated Development Environment – http://www.eclipse.org www.java.sun.com www.java.sun.com/docs/books/tutorial/java/index.html GridWorld Case Study - www.apcentral.collegeboard.com 			<p>Instructional Adjustments: Modifications, student difficulties, possible misunderstandings</p> <ul style="list-style-type: none"> Circulate around classroom during Lab and offer one-on-one assistance for struggling students 		

Standard Data Structures(Cont.)

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Student will be able to understand and use features of Stacks and Queues.

Essential Questions: How are features of Stacks and Queues used to manage large amounts of data?

Unit Assessment: Programming assignments and chapter test

		Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points	
<p>Create and manipulate information, independently and/or collaboratively, to solve problems and design and develop products. 8.1.12.B.9</p> <ul style="list-style-type: none"> • Ability to store and manage large amounts of data using stacks and queue. 	<ul style="list-style-type: none"> • Implement and use stacks and methods of the Stack Interface including top, pop, peekTop, push and size • Implement and use queues and methods of the Queue interface including front, rear, dequeue, enqueue, peekFront and size 	<ul style="list-style-type: none"> • To understand features of a stack such as FIFO (first in, first out) • To understand features of a queues such as LIFO (last in, first out) • Applications of stacks and queues such as parsing expressing in infix and postfix order 	<ul style="list-style-type: none"> • Use PowerPoint slides provided with Teacher Resources • Select several programming projects from the end of the chapter • Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program. 	<ul style="list-style-type: none"> • Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. • Continually emphasize grading exam standards developed by College Board. 	
<p>Resources: Essential Materials, Supplementary Materials, Links to Best Practices</p> <ul style="list-style-type: none"> • Textbook and related Teacher Resources. • Computer Fundamentals - http://www.adita.com/one-a.htm • Integrated Development Environment – http://www.eclipse.org • www.java.sun.com • www.java.sun.com/docs/books/tutorial/java/index.html • GridWorld Case Study - www.apcentral.collegeboard.com 			<p>Instructional Adjustments: Modifications, student difficulties, possible misunderstandings</p> <ul style="list-style-type: none"> • Circulate around classroom during Lab and offer one-on-one assistance for struggling students 		

Standard Data Structures(Cont.)

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Student will be able to understand and use features of Binary Trees and Priority Queues.

Essential Questions: How are features of Binary Trees and Priority Queues used to manage large amounts of data?

Unit Assessment: Programming assignments and chapter test

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points
<p>Create and manipulate information, independently and/or collaboratively, to solve problems and design and develop products. 8.1.12.B.9</p> <ul style="list-style-type: none"> • Ability to store and manage large amounts of data using binary trees and priority queues. 	<ul style="list-style-type: none"> • Implement and use binary trees • Define a full vs. a complete tree • Define the height of a tree • Apply recursive methods to binary trees • Implement and use a binary search tree • Implement and use the heap sort • Traverse a binary tree using preorder, inorder, post order and level order traversal. • Maintain a tree using insertion and removal methods. 	<ul style="list-style-type: none"> • Understand features of a binary tree and the nodes that make up a binary tree • Understand the different ways to traverse a tree. • Understand how a sub-tree is used as the building block of a heap sort 	<ul style="list-style-type: none"> • Use PowerPoint slides provided with Teacher Resources • Select several programming projects from the end of the chapter • Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program. 	<ul style="list-style-type: none"> • Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. • Continually emphasize grading exam standards developed by College Board.

Standard Data Structures(Cont.)**Resources:** Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and related Teacher Resources.
- Computer Fundamentals - <http://www.adita.com/one-a.htm>
- Integrated Development Environment – <http://www.eclipse.org>
- www.java.sun.com
- www.java.sun.com/docs/books/tutorial/java/index.html
- GridWorld Case Study - www.apcentral.collegeboard.com

Instructional Adjustments: Modifications, student difficulties, possible misunderstandings

- Circulate around classroom during Lab and offer one-on-one assistance for struggling students

Standard Algorithms

Targeted Standards: Standard 8.1: 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.

Unit Objectives/Enduring Understandings: Student will be able to understand the difference between recursive and iterative solutions, complexity analysis, implementation of recursive methods.

Essential Questions: When is it efficient to use a recursive method vs. an iterative method when managing large amounts of data and how are these methods labeled?

Unit Assessment: Programming assignments and chapter test

		Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts <i>What students will know.</i>	Skills <i>What students will be able to do.</i>	Activities/Strategies Technology Implementation/ Interdisciplinary Connections	Assessment Check Points	
Make informed choices among technology systems, resources, and services in a variety of contexts.8.1.12.B.3	<ul style="list-style-type: none"> Define and trace through a Recursive method. Recognize a stack overflow error Define and label methods based on their efficiency such as $O(n)$, $O(1)$, $O(n^2)$, $O(n^3)$, $O(\log n)$ Recognize Best Case, Worst Case and Average Case Recognize the Complexity analysis of Quick Sort, Merge Sort, Heap Sort, Selection Sort and Insertion Sort 	<ul style="list-style-type: none"> Identify and label the efficiency of a variety of sorts and recursive methods. 	<ul style="list-style-type: none"> Use PowerPoint slides provided with Teacher Resources Select several programming projects from the end of the chapter Instructor may write programs and allow students to determine the output. Likewise, instructor may supply input and desired output and ask students to write a program. 	<ul style="list-style-type: none"> Use Self-Review questions, Multiple Choice questions, True/False statements and Short Answer questions at the end of chapter. Continually emphasize grading exam standards developed by College Board. 	
<p>Resources: Essential Materials, Supplementary Materials, Links to Best Practices</p> <ul style="list-style-type: none"> Textbook and related Teacher Resources. Computer Fundamentals - http://www.adita.com/one-a.htm Integrated Development Environment – http://www.eclipse.org www.java.sun.com www.java.sun.com/docs/books/tutorial/java/index.html GridWorld Case Study - www.apcentral.collegeboard.com 			<p>Instructional Adjustments: Modifications, student difficulties, possible misunderstandings</p> <ul style="list-style-type: none"> Circulate around classroom during Lab and offer one-on-one assistance for struggling students 		

**Public Schools of Edison Township
Divisions of Curriculum and Instruction**

Draft 14

Essential Instructional Behaviors

Edison's Essential Instructional Behaviors are a collaboratively developed statement of effective teaching from pre-school through Grade 12. This statement of instructional expectations is intended as a framework and overall guide for teachers, supervisors, and administrators; its use as an observation checklist is inappropriate.

1. Planning which Sets the Stage for Learning and Assessment

Does the planning show evidence of:

- a. units and lessons directly related to learner needs, the written curriculum, the New Jersey Core Content Curriculum Standards (NJCCCS), and the Cumulative Progress Indicators (CPI)?
- b. measurable objectives that are based on diagnosis of learner needs and readiness levels and reflective of the written curriculum, the NJCCCS, and the CPI?
- c. lesson design sequenced to make meaningful connections to overarching concepts and essential questions?
- d. provision for effective use of available materials, technology and outside resources?
- e. accurate knowledge of subject matter?
- f. multiple means of formative and summative assessment, including performance assessment, that are authentic in nature and realistically measure learner understanding?
- g. differentiation of instructional content, processes and/or products reflecting differences in learner interests, readiness levels, and learning styles?
- h. provision for classroom furniture and physical resources to be arranged in a way that supports student interaction, lesson objectives, and learning activities?

2. Observed Learner Behavior that Leads to Student Achievement

Does the lesson show evidence of:

- a. learners actively engaged throughout the lesson in on-task learning activities?
- b. learners engaged in authentic learning activities that support reading such as read alouds, guided reading, and independent reading utilizing active reading strategies to deepen comprehension (for example inferencing, predicting, analyzing, and critiquing)?
- c. learners engaged in authentic learning activities that promote writing such as journals, learning logs, creative pieces, letters, charts, notes, graphic organizers and research reports that connect to and extend learning in the content area?
- d. learners engaged in authentic learning activities that promote listening, speaking, viewing skills and strategies to understand and interpret audio and visual media?
- e. learners engaged in a variety of grouping strategies including individual conferences with the teacher, learning partners, cooperative learning structures, and whole-class discussion?
- f. learners actively processing the lesson content through closure activities throughout the lesson?
- g. learners connecting lesson content to their prior knowledge, interests, and personal lives?
- h. learners demonstrating increasingly complex levels of understanding as evidenced through their growing perspective, empathy, and self-knowledge as they relate to the academic content?
- i. learners developing their own voice and increasing independence and responsibility for their learning?
- j. learners receiving appropriate modifications and accommodations to support their learning?

3. Reflective Teaching which Informs Instruction and Lesson Design

Does the instruction show evidence of:

- a. differentiation to meet the needs of all learners, including those with Individualized Education Plans?
- b. modification of content, strategies, materials and assessment based on the interest and immediate needs of students during the lesson?
- c. formative assessment of the learning before, during, and after the lesson, to provide timely feedback to learners and adjust instruction accordingly?
- d. the use of formative assessment by both teacher and student to make decisions about what actions to take to promote further learning?
- e. use of strategies for concept building including inductive learning, discovery-learning and inquiry activities?
- f. use of prior knowledge to build background information through such strategies as anticipatory set, K-W-L, and prediction brainstorm?
- g. deliberate teacher modeling of effective thinking and learning strategies during the lesson?
- h. understanding of current research on how the brain takes in and processes information and how that information can be used to enhance instruction?
- i. awareness of the preferred informational processing strategies of learners who are technologically sophisticated and the use of appropriate strategies to engage them and assist their learning?
- j. activities that address the visual, auditory, and kinesthetic learning modalities of learners?
- k. use of questioning strategies that promote discussion, problem solving, and higher levels of thinking?
- l. use of graphic organizers and hands-on manipulatives?
- m. creation of an environment which is learner-centered, content rich, and reflective of learner efforts in which children feel free to take risks and learn by trial and error?
- n. development of a climate of mutual respect in the classroom, one that is considerate of and addresses differences in culture, race, gender, and readiness levels?
- o. transmission of proactive rules and routines which students have internalized and effective use of relationship-preserving desists when students break rules or fail to follow procedures?

4. Responsibilities and Characteristics which Help Define the Profession

Does the teacher show evidence of:

- a. continuing the pursuit of knowledge of subject matter and current research on effective practices in teaching and learning, particularly as they tie into changes in culture and technology?
- b. maintaining accurate records and completing forms/reports in a timely manner?
- c. communicating with parents about their child's progress and the instructional process?
- d. treating learners with care, fairness, and respect?
- e. working collaboratively and cooperatively with colleagues and other school personnel?
- f. presenting a professional demeanor?