



Springfield Public Schools
Springfield, New Jersey 07081

Advanced Computer Science – Data Structures: Java
Grades 11-12

Curriculum Guide

2022

This curriculum may be modified through varying materials, techniques, and strategies as per an individual student's IEP.

Springfield Board of Education
Approved July 25, 2022



SPRINGFIELD PUBLIC SCHOOLS
ENGAGING STUDENTS • FOSTERING ACHIEVEMENT • CULTIVATING 21ST CENTURY GLOBAL SKILLS

Springfield Public Schools Springfield, New Jersey

**Dr. Rachel Goldberg
Superintendent of Schools**

**Erica Scudero
Assistant Superintendent**

Curriculum Developed By:

Dr. Norman Francis ~ Principal of Jonathan Dayton High School

Gregory Salmon ~ STEM Supervisor

Mary Ellen Powers ~ Math and Computer Science Teacher

Springfield Board of Education
PO Box 210
Springfield, New Jersey 07081

"Cultivating compassionate and extraordinary learners"

Rachel Goldberg
Superintendent of Schools
(973) 376-1025

Michelle Calas
School Business Administrator/
Board Secretary
(973) 376-1025

Resolution: School Government
Introduced By: Mr. Miller
Seconded By: Mrs. Gamarekian
Adopted: July 25, 2022

Curriculum Adoption

To adopt the following curriculum and courses for the 2022-2023 school year:

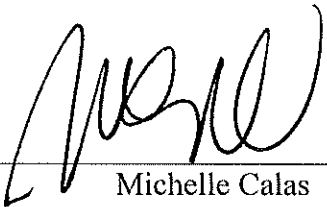
- Revised Elementary Curriculum due to revised state standards.
 - K-5 Social Studies
- Revised Middle School Curriculum due to revised state standards.
 - Grades 6-8 Social Studies
- Revised High School Curriculum due to revised state standards.
 - AP Macroeconomics
 - AP European History
 - AP US Government & Politics
 - AP US History
 - Contemporary Global Studies
 - Diversity Studies
 - Holocaust & Genocide
 - Sociology
 - US History to 1900
 - World History
 - World History Honors
 - 20th Century & Beyond
- Revised Middle School Courses due to new standards in Computer Science
 - PLTW Gateway: Computer Science for Innovators and Makers
 - PLTW Gateway: App Creators
- Revised High School Courses due to new standards in Computer Science
 - Introduction to Java
 - AP Computer Science Principles
 - Advanced Computer Science - Data Structures: Java
- New Elementary Courses
 - Kindergarten Computer Science
 - Grade 1 Computer Science
 - Grade 2 Computer Science
 - Anticipated - moving PLTW Launch down to Grade 3
 - PLTW Launch Grade 4
 - PLTW Launch Grade 5
- New Middle School Elective Course
 - Forensics

- New Elective Courses at FMG
 - Creative Writing
 - Drama/Theater/Playwriting
 - Journalism
 - Mock Trial
 - Model UN
 - Mysteries in History
 - Project Citizen
- New ELA Courses at FMG
 - Accelerated Communication Arts Grade 7
 - Accelerated Communication Arts Grade 8
- New Elective Courses at JDHS:
 - African American Studies

I, Michelle Calas, School Business Administrator/Board Secretary of the Springfield Board of Education, hereby certify that the extract from the minutes of a meeting of the Board of Education duly called and held on July 25, 2022 has been compared by me with the original minute book of said Board of Education, and is a true, complete and exact copy therefor, as far as the same relates to the subject matter referred to said contract.

IN WITNESS THEREOF, I have hereunto set my hand and seal of the Springfield Board of Education, this 26th of July 2022.

SEAL



Michelle Calas
School Business Administrator/Board Secretary



Unit Overview Template

Content Area: Computer Science **Grade Level:** 11-12 **Unit Title:** Lists, Stacks, and Queues

Unit Summary: Students will be able to create lists, stacks, and queues; store, modify, and retrieve data; and determine the appropriate data type to use for various scenarios.

Primary Interdisciplinary Connections: *Mathematics*

21st Century Themes ☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy
☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy

Unit Rationale: In this unit, students will learn about virtual storage and retrieval of data through the use of lists, stacks, and queues.

Career Readiness (Standard 9):

- 9.2.12.CAP.3: Investigate how continuing education contributes to one's career and personal growth.
- 9.2.12.CAP.5: Assess and modify a personal plan to support current interests and postsecondary plans.
- 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.CT.2 - Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.5.CT.3 - Describe how digital tools and technology may be used to solve problems.
- 9.4.12.TL.1 - Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
- 9.4.12.TL.3 - Analyze the effectiveness of the process and quality of collaborative environments.
- 9.4.2.TL.6 - Illustrate and communicate ideas and stories using multiple digital tools (e.g., SL.2.5.).

Learning Targets

Standards: 8.1.12; A-SSE; F-BF

Content Statements: Computing Systems; Data & Analysis; Algorithms & Programming; Interpret the structure of expressions; Build a function that models a relationship between two quantities

CPI#:	Cumulative Progress Indicator (CPI):
8.1.12.CS.1	Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
8.1.12.CS.2	Model interactions between application software, system software, and hardware.
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.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.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.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.
A-SSE.A.1	Interpret expressions that represent a quantity in terms of its context. a. Interpret parts of an expression, such as terms, factors, and coefficients. b. Interpret complicated expressions by viewing one or more of their parts as a single entity. For example, interpret $P(1+r)^n$ as the product of P and a factor not depending on P .
A-SSE.A.2	Use the structure of an expression to identify ways to rewrite it. For example, see $x^4 - y^4$ as $(x^2)^2 - (y^2)^2$, thus recognizing it as a difference of squares that can be factored as $(x^2 - y^2)(x^2 + y^2)$.
F-BF.A.1	Write a function that describes a relationship between two quantities. a. Determine an explicit expression, a recursive process, or steps for calculation from a context. b. Combine standard function types using arithmetic operations. For example, build a function that models the temperature of a cooling body by adding a constant function to a decaying exponential, and relate these functions to the model. c. (+) Compose functions. For example, if $T(y)$ is the temperature in the atmosphere as a function of height, and $h(t)$ is the height of a weather balloon as a function of time, then $T(h(t))$ is the temperature at the location of the weather balloon as a function of time.
F-BF.A.2	Write arithmetic and geometric sequences both recursively and with an explicit formula, use them to model situations, and translate between the two forms.
SEL.PK-12.2.2	Recognize the skills needed to establish and achieve personal and educational goals.
SEL.PK-12.4.1	Develop, implement and model effective problem-solving, and critical thinking skills.
SEL.PK-12.5.2	Utilize positive communication and social skills to interact effectively with others.

Unit Essential Questions:

- What is an algorithm?
- What are data Structures?
- Why is understanding of data structures important?
- What are different types of data structures?



- What is the difference between primitive and non primitive data structures?

Unit Enduring Understandings:

- Information can be represented as digital data
- Data structures are efficient building blocks for software
- A data structure defines the way data elements can be stored and retrieved
- Data structures defined the way program elements are organized

Unit Learning Targets:

Students will....create lists, stacks, and queues; store, modify, and retrieve data; and determine the appropriate data type to use for various scenarios.

Evidence of Learning

Summative Assessment: Unit Test

Equipment Needed: Laptops

Teacher Resources: Laptop and projector

Formative Assessments:

- Student practice worksheets for pseudocode and syntax
- Student practice programs
- Individual programming editing conferences with students
- Students demonstrate programs
- Quizzes



Lesson Plans

Lesson	Time Frame
Lesson 1 Lists	2 days
Lesson 2 Stacks	2 days
Lesson 3 Queues	2 days

Teacher Notes:

Ensure students are looking for opportunities to optimize their code.

Curriculum Development Resources/Technology Resources:

Click the links below to access additional resources used to design this unit:

<https://www.greenfoot.org/door>

<https://alice.org>

<https://stackoverflow.com/>

<https://www.w3schools.in/data-structures/intro>

<https://www.w3schools.com/java/default.asp>

<https://www.learnjavaonline.org/>



Lesson Plan 1 Template

Content Area: Computer Science Lesson Title: Lists Time Frame: 2 days

Lesson Components

21st Century Themes

- ☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy
☐ Civic Literacy ☐ Health Literacy

21st Century Skills

- ☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy
☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy

Interdisciplinary Connections:

Mathematics

Integration of Technology:

Computer programming

Equipment Needed:

Laptops

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: • create lists. • insert data in lists. • delete data from lists. • replace data in lists.	Lesson Sequence 1. Students will recall how to create objects in Java. The teacher then works with the students to create lists. 2. The students complete guided practice on inserting, deleting, and replacing data in lists. 3. The students then work on creating lists of items in an imaginary store. 4. The class comes back together to discuss their code, and makes suggestions to each other.	• Student practice programs • Programming editing conferences with student • Students design solutions • Students test and demonstrate code
Differentiation: Programming assignments will be modified for students as needed.		
Resources Provided: • Laptop • Java programming language • Java index of source code libraries		



Lesson Plan 2 Template

Content Area: Computer Science Lesson Title: Stacks Time Frame: 2 days

Lesson Components

21st Century Themes

- ☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy
☐ Civic Literacy ☐ Health Literacy

21st Century Skills

- ☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy
☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy

Interdisciplinary Connections:
 Mathematics

Integration of Technology:
 Computer programming

Equipment Needed:
 Laptops

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: • create stacks. • insert data in a stack. • delete data from stacks. • replace data in a stack. • merge stored data from multiple stacks into a single stack	Lesson Sequence 1. Students will recall how to create lists in Java. The teacher then works with the students to create stacks. 2. The students complete guided practice on inserting, deleting, and replacing data in stacks. 3. The students then work on creating stacks of numerical prices of the items from an imaginary store. 4. The class comes back together to discuss their code, and makes suggestions to each other.	• Student practice programs • Programming editing conferences with student • Students design solutions • Students test and demonstrate code
Differentiation: Programming assignments will be modified for students as needed.		
Resources Provided: • Laptop • Java programming language • Java index of source code libraries		



Lesson Plan 3 Template

Content Area: Computer Science Lesson Title: Queues Time Frame: 2 days

Lesson Components

21st Century Themes

- ☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy
☐ Civic Literacy ☐ Health Literacy

21st Century Skills

- ☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy
☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy

Interdisciplinary Connections:

Mathematics

Integration of Technology:

Computer programming

Equipment Needed:

Laptops

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: • create queues. • insert data into a queue. • delete data from a queue. • replace data in a queue. • merge data from multiple queues into a single queue.	Lesson Sequence 1. Students will recall how to create stacks in Java. The teacher then works with the students to create queues. 2. The students complete guided practice on inserting, deleting, and replacing data in queues. 3. The students then work on creating queues of items in an imaginary store as they are purchased, noting the first-in-first-out nature of the data. 4. The class comes back together to discuss their code, and makes suggestions to each other.	• Student practice programs • Programming editing conferences with student • Students design solutions • Students test and demonstrate code
Differentiation: Programming assignments will be modified for students as needed.		
Resources Provided: • Laptop • Java programming language • Java index of source code libraries		



Lesson Reflection

Reflect on the lesson you have developed and rate the degree to which the lesson *Strongly*, *Moderately* or *Weakly* meets the criteria below.

Lesson Activities	Strongly	Moderately	Weakly
Are challenging Strongly and require higher order thinking and problem solving skills	x		
Allow for student choice		x	
Provide scaffolding for acquiring targeted knowledge/skills	x		
Integrate global perspectives	x		
Integrate 21st century skills	x		
Provide opportunities for interdisciplinary connection and transfer of knowledge and skills	x		
Foster student use of technology as a tool to develop critical thinking, creativity and innovation skills	x		
Are varied to address different student learning styles and preferences		x	
Are differentiated based on student needs		x	
Are student-centered with teacher acting as a facilitator and co-learner during the teaching and learning process	x		
Provide means for students to demonstrate knowledge and skills and progress in meeting learning goals and objectives	x		
Provide opportunities for student reflection and self- assessment	x		
Provide data to inform and adjust instruction to better meet the varying needs of learners	x		

Curriculum Design Template

Content Area: Computer Science	Grade Level: 11-12	Unit Title: Lists, Stacks, and Queues
Created: June 2022	Board Approved on:	



Unit Overview Template

Content Area:	Computer Science	Grade Level:	11-12	Unit Title:	Linked Lists
Unit Summary: Students will work with data storage and retrieval from singly linked lists, doubly linked lists, and circular linked lists. Linked lists will be analyzed as a data structure made up of a chain of nodes					
Primary Interdisciplinary Connections: <i>Mathematics</i>					
21st Century Themes ☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy ☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy					
Unit Rationale: Linked lists provide a basis for efficient insertion and deletion of data structures, such as stacks, queues, and arrays.					
Career Readiness (Standard 9): 9.2.12.CAP.3: Investigate how continuing education contributes to one's career and personal growth. 9.2.12.CAP.5: Assess and modify a personal plan to support current interests and postsecondary plans. 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.CT.2 - Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a). 9.4.5.CT.3 - Describe how digital tools and technology may be used to solve problems. 9.4.12.TL.1 - Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.). 9.4.12.TL.3 - Analyze the effectiveness of the process and quality of collaborative environments. 9.4.2.TL.6 - Illustrate and communicate ideas and stories using multiple digital tools (e.g., SL.2.5.).					

Learning Targets

Standards: 8.1.12; A-SSE; F-BF	
Content Statements: Computing Systems; Data & Analysis; Algorithms & Programming; Interpret the structure of expressions; Build a function that models a relationship between two quantities	
CPI#:	Cumulative Progress Indicator (CPI):
8.1.12.CS.1	Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
8.1.12.CS.2	Model interactions between application software, system software, and hardware.
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.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.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.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.
A-SSE.A.1	Interpret expressions that represent a quantity in terms of its context. a. Interpret parts of an expression, such as terms, factors, and coefficients. b. Interpret complicated expressions by viewing one or more of their parts as a single entity. For example, interpret $P(1+r)^n$ as the product of P and a factor not depending on P .
A-SSE.A.2	Use the structure of an expression to identify ways to rewrite it. For example, see $x^4 - y^4$ as $(x^2)^2 - (y^2)^2$, thus recognizing it as a difference of squares that can be factored as $(x^2 - y^2)(x^2 + y^2)$.
F-BF.A.1	Write a function that describes a relationship between two quantities. a. Determine an explicit expression, a recursive process, or steps for calculation from a context. b. Combine standard function types using arithmetic operations. For example, build a function that models the temperature of a cooling body by adding a constant function to a decaying exponential, and relate these functions to the model. c. (+) Compose functions. For example, if $T(y)$ is the temperature in the atmosphere as a function of height, and $h(t)$ is the height of a weather balloon as a function of time, then $T(h(t))$ is the temperature at the location of the weather balloon as a function of time.
F-BF.A.2	Write arithmetic and geometric sequences both recursively and with an explicit formula, use them to model situations, and translate between the two forms.
SEL.PK-12. 2.2	Recognize the skills needed to establish and achieve personal and educational goals.
SEL.PK-12. 4.1	Develop, implement and model effective problem-solving, and critical thinking skills.
SEL.PK-12. 5.2	Utilize positive communication and social skills to interact effectively with others.

Unit Essential Questions:

- What are linked lists?



- What are different types of linked lists?
- How do you create and work with singly, doubly, and circular linked lists?
- How do you delete and insert data into singly, doubly, and circular linked lists?
- How does this compare to an array list?

Unit Enduring Understandings:

- Singly, doubly, and circular linked lists are used to implement abstract data types.
- Operations using linked lists.
- Comparison between linked lists and array lists.

Unit Learning Targets:

Students will....analyze and apply linked lists to a variety of situations.

Evidence of Learning

Summative Assessment: Unit Test

Equipment Needed: Laptops

Teacher Resources: Laptop and projector

Formative Assessments:

- Student practice worksheets for pseudocode and syntax
- Student practice programs
- Individual programming editing conferences with students
- Students demonstrate programs
- Quizzes



Lesson Plans

Lesson	Time Frame
Lesson 1 Singly Linked Lists	2 days
Lesson 2 Doubly Linked Lists	2 days
Lesson 3 Circular Linked Lists	2 days

Teacher Notes:

Ensure students are looking for opportunities to optimize their code.

Curriculum Development Resources/Technology Resources:

Click the links below to access additional resources used to design this unit:

<https://www.greenfoot.org/door>

<https://alice.org>

<https://stackoverflow.com/>

<https://www.w3schools.in/data-structures/intro>

<https://www.w3schools.com/java/default.asp>

<https://www.learnjavaonline.org/>



Lesson Plan 1 Template

Content Area: Computer Science Lesson Title: Singly Linked Lists Time Frame: 2 days

Lesson Components

21st Century Themes

- ☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy
☐ Civic Literacy ☐ Health Literacy

21st Century Skills

- ☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy
☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy

Interdisciplinary Connections:

Mathematics

Integration of Technology:

Computer programming

Equipment Needed:

Laptops

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: • create singly linked lists. • insert data in singly linked lists. • delete data from singly linked lists. • replace data in singly linked lists. • access data in singly linked lists.	Lesson Sequence 1. Students will model a list on paper, then work to build the understanding of linked lists. 2. The class will work together to analyze a singly linked list, noting the single direction the link occurs. 3. Students will work on guided practice to insert, delete, replace, and access data in a given singly linked list. 4. Students then write their own singly linked lists in code. 5. The teacher observes the class and meets with	• Student practice programs • Programming editing conferences with student • Students design solutions • Students test and demonstrate code



	individuals for editing conferences.	
--	--------------------------------------	--

Differentiation:

Programming assignments will be modified for students as needed.

Resources Provided:

- Laptop
- Java programming language
- Java index of source code libraries



Lesson Plan 2 Template		
Content Area: Computer Science	Lesson Title: Doubly Linked Lists	Time Frame: 2 days
Lesson Components		
<u>21st Century Themes</u>		
☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy ☐ Civic Literacy ☐ Health Literacy		
<u>21st Century Skills</u>		
☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy ☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy		
Interdisciplinary Connections: Mathematics		
Integration of Technology: Computer programming		
Equipment Needed: Laptops		

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: • create doubly linked lists. • insert data in doubly linked lists. • delete data from doubly linked lists. • replace data in doubly linked lists. • access data in doubly linked lists.	Lesson Sequence 1. The students will pair up and review singly linked lists. 2. The class will work together to analyze a doubly linked list, noting the two directions the link occurs. 3. Students will work on guided practice to insert, delete, replace, and access data in a given doubly linked list. 4. Students then write their own doubly linked lists in code. 5. The teacher observes the class and meets with	• Student practice programs • Programming editing conferences with student • Students design solutions • Students test and demonstrate code



	individuals for editing conferences.	
--	--------------------------------------	--

Differentiation:

Programming assignments will be modified for students as needed.

Resources Provided:

- Laptop
- Java programming language
- Java index of source code libraries



Lesson Plan 3 Template		
Content Area: Computer Science	Lesson Title: Circular Linked Lists	Time Frame: 2 days
Lesson Components		
<u>21st Century Themes</u>		
☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy ☐ Civic Literacy ☐ Health Literacy		
<u>21st Century Skills</u>		
☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy ☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy		
Interdisciplinary Connections: Mathematics		
Integration of Technology: Computer programming		
Equipment Needed: Laptops		

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: • create circular linked lists. • insert data in circular linked lists. • delete data from circular linked lists. • replace data in circular linked lists. • access data in circular linked lists.	Lesson Sequence 1. Students will pair up and review singly and doubly linked lists, noting any similarities. Students are then asked “What can we add to these lists but still maintain link direction.” 2. The class will work together to analyze a circular singly linked list, noting what happens at the end of the list. 3. The class will work together to analyze a circular doubly linked list, noting what happens at the end of the list. 4. Students will work on guided practice to insert,	• Student practice programs • Programming editing conferences with student • Students design solutions • Students test and demonstrate code



delete, replace, and
access data in a given
circular linked list.

5. Students then write their
own circular linked lists
in code.
6. The teacher observes the
class and meets with
individuals for editing
conferences.

Differentiation:

Programming assignments will be modified for students as needed.

Resources Provided:

- Laptop
- Java programming language
- Java index of source code libraries



Lesson Reflection

Reflect on the lesson you have developed and rate the degree to which the lesson *Strongly*, *Moderately* or *Weakly* meets the criteria below.

Lesson Activities	Strongly	Moderately	Weakly
Are challenging Strongly and require higher order thinking and problem solving skills	X		
Allow for student choice		X	
Provide scaffolding for acquiring targeted knowledge/skills	X		
Integrate global perspectives	X		
Integrate 21st century skills	X		
Provide opportunities for interdisciplinary connection and transfer of knowledge and skills	X		
Foster student use of technology as a tool to develop critical thinking, creativity and innovation skills	X		
Are varied to address different student learning styles and preferences		X	
Are differentiated based on student needs		X	
Are student-centered with teacher acting as a facilitator and co-learner during the teaching and learning process	X		
Provide means for students to demonstrate knowledge and skills and progress in meeting learning goals and objectives	X		
Provide opportunities for student reflection and self- assessment	X		
Provide data to inform and adjust instruction to better meet the varying needs of learners	X		

Curriculum Design Template

Content Area: Computer Science	Grade Level: 11-12	Unit Title: Linked Lists
Created: June 2022	Board Approved on:	



Unit Overview Template

Content Area:	Computer Science	Grade Level:	11-12	Unit Title:	Hashtables & Hashmaps
----------------------	------------------	---------------------	-------	--------------------	-----------------------

Unit Summary: Students will use Hashtables to store an association between a value and its key. Students will apply this to build dictionaries and phonebooks. Students will use Hashmaps to map keys to their associated values.

Primary Interdisciplinary Connections: *Mathematics, Technology, 21st Century*

21st Century Themes ☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy
☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy

Unit Rationale: Hashtables and Hashmaps are used when associating values to keys.

Career Readiness (Standard 9):

- 9.2.12.CAP.3: Investigate how continuing education contributes to one's career and personal growth.
 9.2.12.CAP.5: Assess and modify a personal plan to support current interests and postsecondary plans.
 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.CT.2 - Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
 9.4.5.CT.3 - Describe how digital tools and technology may be used to solve problems.
 9.4.12.TL.1 - Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
 9.4.12.TL.3 - Analyze the effectiveness of the process and quality of collaborative environments.
 9.4.2.TL.6 - Illustrate and communicate ideas and stories using multiple digital tools (e.g., SL.2.5.).

Learning Targets

Standards: 8.1.12; A-SSE; F-BF	
Content Statements: Computing Systems; Data & Analysis; Algorithms & Programming; Interpret the structure of expressions; Build a function that models a relationship between two quantities	
CPI#:	Cumulative Progress Indicator (CPI):
8.1.12.CS.1	Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
8.1.12.CS.2	Model interactions between application software, system software, and hardware.
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.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.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.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.
A-SSE.A.1	Interpret expressions that represent a quantity in terms of its context. a. Interpret parts of an expression, such as terms, factors, and coefficients. b. Interpret complicated expressions by viewing one or more of their parts as a single entity. For example, interpret $P(1+r)^n$ as the product of P and a factor not depending on P .
A-SSE.A.2	Use the structure of an expression to identify ways to rewrite it. For example, see $x^4 - y^4$ as $(x^2)^2 - (y^2)^2$, thus recognizing it as a difference of squares that can be factored as $(x^2 - y^2)(x^2 + y^2)$.
F-BF.A.1	Write a function that describes a relationship between two quantities. a. Determine an explicit expression, a recursive process, or steps for calculation from a context. b. Combine standard function types using arithmetic operations. For example, build a function that models the temperature of a cooling body by adding a constant function to a decaying exponential, and relate these functions to the model. c. (+) Compose functions. For example, if $T(y)$ is the temperature in the atmosphere as a function of height, and $h(t)$ is the height of a weather balloon as a function of time, then $T(h(t))$ is the temperature at the location of the weather balloon as a function of time.
F-BF.A.2	Write arithmetic and geometric sequences both recursively and with an explicit formula, use them to model situations, and translate between the two forms.
SEL.PK-12.2.2	Recognize the skills needed to establish and achieve personal and educational goals.
SEL.PK-12.4.1	Develop, implement and model effective problem-solving, and critical thinking skills.
SEL.PK-12.5.2	Utilize positive communication and social skills to interact effectively with others.

Unit Essential Questions:

- How do we apply Hashtables?
- How do we apply Hashmaps?
- Why is a Hashmap faster to use than a Hashtable?



Unit Enduring Understandings:

- Use transversals with a Hashtable.
- Insert data in a Hashtable.
- Use transversals with a Hashmap.
- Insert data in a Hashmap.

Unit Learning Targets:

Students will....analyze and apply Hashtables and Hashmaps to data structures.

Evidence of Learning

Summative Assessment: Unit Test

Equipment Needed: Laptops

Teacher Resources: Laptop and projector

Formative Assessments:

- Student practice worksheets for pseudocode and syntax
- Student practice programs
- Individual programming editing conferences with students
- Students demonstrate programs
- Quizzes



Lesson Plans

Lesson	Time Frame
Lesson 1 Hashtables	2 days
Lesson 2 Hashmaps	2 days
Lesson 3 Testing Boundary Cases	2 days

Teacher Notes:

Ensure students are looking for opportunities to optimize their code.

Curriculum Development Resources/Technology Resources:

Click the links below to access additional resources used to design this unit:

<https://www.greenfoot.org/door>

<https://alice.org>

<https://stackoverflow.com/>

<https://www.w3schools.in/data-structures/intro>

<https://www.w3schools.com/java/default.asp>

<https://www.learnjavaonline.org/>



Lesson Plan 1 Template

Content Area: Computer Science Lesson Title: Hashtables Time Frame: 2 days

Lesson Components

21st Century Themes

- ☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy
☐ Civic Literacy ☐ Health Literacy

21st Century Skills

- ☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy
☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy

Interdisciplinary Connections:
 Mathematics

Integration of Technology:
 Computer programming

Equipment Needed:
 Laptops

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: • use traversals with Hashtables. • insert data into a Hashtable. • delete data from a Hashtable.	Lesson Sequence 1. The class reviews what an array is. 2. Students lead the discussion to determine the link between arrays and Hashtables. 3. Analyze the Big O number for an algorithm, then compare to the Big O number of a similar algorithm rewritten using Hashtables. 4. Students work in pairs on guided practice applying Hashtables to their code. Students should build a program that checks for keys. 5. End with a class discussion on security of	• Student practice programs • Programming editing conferences with student • Students design solutions • Students test and demonstrate code



	keys when working with Hashtables.	
--	---------------------------------------	--

Differentiation:

Programming assignments will be modified for students as needed.

Resources Provided:

- Laptop
- Java programming language
- Java index of source code libraries



Lesson Plan 2 Template		
Content Area: Computer Science	Lesson Title: Hashmaps	Time Frame: 2 days
Lesson Components		
<u>21st Century Themes</u>		
☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy ☐ Civic Literacy ☐ Health Literacy		
<u>21st Century Skills</u>		
☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy ☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy		
Interdisciplinary Connections: Mathematics		
Integration of Technology: Computer programming		
Equipment Needed: Laptops		

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: • use traversals with Hashmaps. • insert data into a Hashmap. • delete data from a Hashmap.	Lesson Sequence 1. The class reviews what a Hashtable is. 2. Students lead the discussion to determine the link between Hashtables and Hashmaps. 3. Analyze the Big O number for an algorithm, then compare to the Big O number of a similar algorithm rewritten using Hashmaps. 4. Students work in pairs on guided practice applying Hashmaps to their code. Students should build a program that checks for keys.	• Student practice programs • Programming editing conferences with student • Students design solutions • Students test and demonstrate code



- | | | |
|--|--|--|
| | 5. End with a class discussion on security of keys when working with Hashmaps. | |
|--|--|--|

Differentiation:

Programming assignments will be modified for students as needed.

Resources Provided:

- Laptop
- Java programming language
- Java index of source code libraries



Lesson Plan 3 Template

Content Area: Computer Science Lesson Title: Testing Boundary Cases Time Frame: 2 days

Lesson Components

21st Century Themes

- ☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy
☐ Civic Literacy ☐ Health Literacy

21st Century Skills

- ☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy
☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy

Interdisciplinary Connections:
 Mathematics

Integration of Technology:
 Computer programming

Equipment Needed:
 Laptops

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: - perform a series of error tests. - determine appropriate algorithms.	Lesson Sequence - Students are given a series of algorithms and code to analyze on their own to test boundary cases. - Students then meet with a partner to discuss error tests of the given code. - Students debug and rewrite the algorithms, then compare Big O numbers with other groups. - The teacher observes the class and meets with individuals for editing conferences.	- Student practice programs - Programming editing conferences with student - Students design solutions - Students test and demonstrate code



Differentiation:

Programming assignments will be modified for students as needed.

Resources Provided:

- Laptop
- Java programming language
- Java index of source code libraries



Lesson Reflection

Reflect on the lesson you have developed and rate the degree to which the lesson *Strongly*, *Moderately* or *Weakly* meets the criteria below.

Lesson Activities	Strongly	Moderately	Weakly
Are challenging Strongly and require higher order thinking and problem solving skills	x		
Allow for student choice		x	
Provide scaffolding for acquiring targeted knowledge/skills	x		
Integrate global perspectives	x		
Integrate 21st century skills	x		
Provide opportunities for interdisciplinary connection and transfer of knowledge and skills	x		
Foster student use of technology as a tool to develop critical thinking, creativity and innovation skills	x		
Are varied to address different student learning styles and preferences		x	
Are differentiated based on student needs		x	
Are student-centered with teacher acting as a facilitator and co-learner during the teaching and learning process	x		
Provide means for students to demonstrate knowledge and skills and progress in meeting learning goals and objectives	x		
Provide opportunities for student reflection and self- assessment	x		
Provide data to inform and adjust instruction to better meet the varying needs of learners	x		

Curriculum Design Template

Content Area: Computer Science	Grade Level: 11-12	Unit Title: Hashtables & Hashmaps
Created: June 2022	Board Approved on:	



Unit Overview Template

Content Area: Computer Science	Grade Level: 11-12	Unit Title: Binary Search Trees and Sorting
---------------------------------------	---------------------------	--

Unit Summary: Students will use Binary Search Trees to test classes and libraries, determine boundary cases, analyze algorithms, and perform error testing. Students will store objects in a Tree Node. Students will then identify and analyze code sorting/searching techniques and algorithms.

Primary Interdisciplinary Connections: *Mathematics, Technology, 21st Century*

21st Century Themes ☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy
☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy

Unit Rationale: Searching and sorting algorithms and techniques are used to optimize code.

Career Readiness (Standard 9):

- 9.2.12.CAP.3: Investigate how continuing education contributes to one's career and personal growth.
- 9.2.12.CAP.5: Assess and modify a personal plan to support current interests and postsecondary plans.
- 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.CT.2 - Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
- 9.4.5.CT.3 - Describe how digital tools and technology may be used to solve problems.
- 9.4.12.TL.1 - Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task (e.g., W.11-12.6.).
- 9.4.12.TL.3 - Analyze the effectiveness of the process and quality of collaborative environments.
- 9.4.2.TL.6 - Illustrate and communicate ideas and stories using multiple digital tools (e.g., SL.2.5.).

Learning Targets

Standards: 8.1.12; A-SSE; F-BF

Content Statements: Computing Systems; Data & Analysis; Algorithms & Programming; Interpret the structure of expressions; Build a function that models a relationship between two quantities

CPI#:	Cumulative Progress Indicator (CPI):
8.1.12.CS.1	Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
8.1.12.CS.2	Model interactions between application software, system software, and hardware.
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.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.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.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.
A-SSE.A.1	Interpret expressions that represent a quantity in terms of its context. a. Interpret parts of an expression, such as terms, factors, and coefficients. b. Interpret complicated expressions by viewing one or more of their parts as a single entity. For example, interpret $P(1+r)^n$ as the product of P and a factor not depending on P .
A-SSE.A.2	Use the structure of an expression to identify ways to rewrite it. For example, see $x^4 - y^4$ as $(x^2)^2 - (y^2)^2$, thus recognizing it as a difference of squares that can be factored as $(x^2 - y^2)(x^2 + y^2)$.
F-BF.A.1	Write a function that describes a relationship between two quantities. a. Determine an explicit expression, a recursive process, or steps for calculation from a context. b. Combine standard function types using arithmetic operations. For example, build a function that models the temperature of a cooling body by adding a constant function to a decaying exponential, and relate these functions to the model. c. (+) Compose functions. For example, if $T(y)$ is the temperature in the atmosphere as a function of height, and $h(t)$ is the height of a weather balloon as a function of time, then $T(h(t))$ is the temperature at the location of the weather balloon as a function of time.
F-BF.A.2	Write arithmetic and geometric sequences both recursively and with an explicit formula, use them to model situations, and translate between the two forms.
SEL.PK-12.2.2	Recognize the skills needed to establish and achieve personal and educational goals.
SEL.PK-12.4.1	Develop, implement and model effective problem-solving, and critical thinking skills.
SEL.PK-12.5.2	Utilize positive communication and social skills to interact effectively with others.

Unit Essential Questions:

- How do you use binary trees?
- What advantages do binary search trees give?
- How do you perform error testing using binary trees?
- How do you analyze sorting and searching algorithms?



- When do you use particular sorting and searching techniques?

Unit Enduring Understandings:

- Applications of binary trees.
- Designing and implementing a binary search tree using tree nodes.
- Storing objects in a tree node.
- Analyzing worst and average case Big O values for sorting algorithms.

Unit Learning Targets:

Students will....apply searching and sorting algorithms and techniques to optimize code.

Evidence of Learning

Summative Assessment: Unit Test

Equipment Needed: Laptops

Teacher Resources: Laptop and projector

Formative Assessments:

- Student practice worksheets for pseudocode and syntax
- Student practice programs
- Individual programming editing conferences with students
- Students demonstrate programs
- Quizzes



Lesson Plans

Lesson	Time Frame
Lesson 1 Binary Search Trees	2 days
Lesson 2 Quicksort	2 days
Lesson 3 Heap Sort	2 days

Teacher Notes:

Ensure students are looking for opportunities to optimize their code.

Curriculum Development Resources/Technology Resources:

Click the links below to access additional resources used to design this unit:

<https://www.greenfoot.org/door>

<https://alice.org>

<https://stackoverflow.com/>

<https://www.w3schools.in/data-structures/intro>

<https://www.w3schools.com/java/default.asp>

<https://www.learnjavaonline.org/>



Lesson Plan 1 Template

Content Area: Computer Science Lesson Title: Binary Search Trees Time Frame: 2 days

Lesson Components

21st Century Themes

- ☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy
☐ Civic Literacy ☐ Health Literacy

21st Century Skills

- ☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy
☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy

Interdisciplinary Connections:
 Mathematics

Integration of Technology:
 Computer programming

Equipment Needed:
 Laptops

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: - use Trees and Binary Trees. - design and implement a Binary Search Tree using Tree Nodes. - store Objects in a Tree Node inside a Binary Search Tree.	Lesson Sequence - Students will work with a sample Binary Search Tree and determine the key-value pairs. - Students will work to reorder key-value pairs. - Students will work on guided practice to design and implement a Binary Search Tree using Tree Nodes. - Students will create a Tree Node inside a Binary Search Tree. - The teacher observes the class and meets with individuals for editing conferences.	- Student practice programs - Programming editing conferences with student - Students design solutions - Students test and demonstrate code
Differentiation: Programming assignments will be modified for students as needed.		
Resources Provided: - Laptop - Java programming language - Java index of source code libraries		



Lesson Plan 2 Template		
Content Area: Computer Science	Lesson Title: Quicksort	Time Frame: 2 days
Lesson Components		
<u>21st Century Themes</u>		
☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy ☐ Civic Literacy ☐ Health Literacy		
<u>21st Century Skills</u>		
☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy ☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy		
Interdisciplinary Connections: Mathematics		
Integration of Technology: Computer programming		
Equipment Needed: Laptops		

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: - identify and analyze code sorting and searching algorithms. - apply quicksort to data. - determine drawbacks to using quicksort.	Lesson Sequence - Students will work with sample code illustrating the quicksort technique. - Students will work to debug any inefficiencies in the code. - Students will work on guided practice to design and implement quicksort. - Students will create a dataset, then apply quicksort. - The teacher observes the class and meets with individuals for editing conferences.	- Student practice programs - Programming editing conferences with student - Students design solutions - Students test and demonstrate code
Differentiation: Programming assignments will be modified for students as needed.		



Resources Provided:

- Laptop
- Java programming language
- Java index of source code libraries



Lesson Plan 3 Template		
Content Area: Computer Science	Lesson Title: Heap Sort	Time Frame: 2 days
Lesson Components		
<u>21st Century Themes</u>		
☒ Global Awareness ☒ Financial, Economic, Business and Entrepreneurial Literacy ☐ Civic Literacy ☐ Health Literacy		
<u>21st Century Skills</u>		
☒ Creativity and Innovation ☒ Critical Thinking and Problem Solving ☐ ICT Literacy ☒ Information Literacy ☒ Life and Career Skills ☒ Communication and Collaboration ☐ Media Literacy		
Interdisciplinary Connections: Mathematics		
Integration of Technology: Computer programming		
Equipment Needed: Laptops		

Goals/Objectives	Learning Activities/ Instructional Strategies	Formative Assessment Tasks
Students will: - identify and analyze code sorting and searching algorithms. - apply heap sort to data. - determine drawbacks to using heap sort. - compare different sorting and searching algorithms.	Lesson Sequence - Students will work with sample code illustrating the heap sort technique. - Students will work to debug any inefficiencies in the code. - Students will work on guided practice to design and implement heap sort. - Students will create a dataset, then apply heap sort. - The teacher observes the class and meets with individuals for editing conferences. - The class then comes together to discuss the	- Student practice programs - Programming editing conferences with student - Students design solutions - Students test and demonstrate code



	benefits of quicksort versus heap sort.	
--	--	--

Differentiation:

Programming assignments will be modified for students as needed.

Resources Provided:

- Laptop
- Java programming language
- Java index of source code libraries



Lesson Reflection

Reflect on the lesson you have developed and rate the degree to which the lesson *Strongly*, *Moderately* or *Weakly* meets the criteria below.

Lesson Activities	Strongly	Moderately	Weakly
Are challenging Strongly and require higher order thinking and problem solving skills	x		
Allow for student choice		x	
Provide scaffolding for acquiring targeted knowledge/skills	x		
Integrate global perspectives	x		
Integrate 21st century skills	x		
Provide opportunities for interdisciplinary connection and transfer of knowledge and skills	x		
Foster student use of technology as a tool to develop critical thinking, creativity and innovation skills	x		
Are varied to address different student learning styles and preferences		x	
Are differentiated based on student needs		x	
Are student-centered with teacher acting as a facilitator and co-learner during the teaching and learning process	x		
Provide means for students to demonstrate knowledge and skills and progress in meeting learning goals and objectives	x		
Provide opportunities for student reflection and self- assessment	x		
Provide data to inform and adjust instruction to better meet the varying needs of learners	x		

Curriculum Design Template

Content Area: Computer Science	Grade Level: 11-12	Unit Title: Binary Search Trees and Sorting
Created: June 2022	Board Approved on:	



SPRINGFIELD PUBLIC SCHOOLS

**ENGAGING STUDENTS • FOSTERING ACHIEVEMENT • CULTIVATING 21ST CENTURY
GLOBAL SKILLS**

IEP AND 504 ACCOMMODATIONS AND MODIFICATIONS FOR EARLY CHILDHOOD AND ELEMENTARY SCHOOL STUDENTS:

Large print textbooks
Textbooks for at-home use
Additional time for assignments
Review of directions
Use of mnemonics
Have student restate information
Concrete examples
Adaptive writing utensils
Support auditory presentations with visuals
Use of a study carrel
Assistance in maintaining uncluttered space
Weekly home-school communication tools (notebook, daily log, phone calls or email messages)
Space for movement or breaks
Study sheets and teacher outlines
Extra visual and verbal cues and prompts
Math sheets with highlighted instructions
Graph paper to assist in organizing or lining up math problems
Use of computers and calculators
Books on tape
Graphic organizers
Quiet corner or room to calm down and relax when anxious
Preferential seating
Alteration of the classroom arrangement
Reduction of distractions
Answers to be dictated
Hands-on activities
Use of manipulatives
No penalty for spelling errors or sloppy handwriting
Follow a routine/schedule
Alternate quiet and active time
Teach time management skills
Rest breaks
Verbal and visual cues regarding directions and staying on task
Agenda book and checklists
Adjusted assignment timelines
Visual daily schedule



SPRINGFIELD PUBLIC SCHOOLS

**ENGAGING STUDENTS • FOSTERING ACHIEVEMENT • CULTIVATING 21ST CENTURY
GLOBAL SKILLS**

Varied reinforcement procedures
Immediate feedback
Work-in-progress check
Personalized examples

TESTING AND ASSESSMENT ACCOMMODATIONS AND MODIFICATIONS:

Answers to be dictated
Frequent rest breaks
Additional time
Oral testing
Untimed tests
Choice of test format (multiple-choice, essay, true-false)
Alternate ways to evaluate (projects or oral presentations instead of written tests)
Accept short answers
Open-book tests
Read test and directions to student
Provide study outline prior to tests
Highlight key directions
Test in alternative site
Use of calculator
Extra credit option
Preview test procedures
Simplified test wording; rephrased test questions and/or directions

GENERAL ACCOMMODATIONS AND MODIFICATIONS:

Use of alternative books or materials on the topic being studied
Computerized spell-check support
Word bank of choices for answers to test questions
Provision of calculator and/or number line for math tests
Reworded questions in simpler language
Projects instead of written reports
Highlighting important words or phrases in reading assignments
Modified workload or length of assignments/tests
Modified time demands
Modified grades based on IEP

BEHAVIOR ACCOMMODATIONS MODIFICATIONS:



SPRINGFIELD PUBLIC SCHOOLS

**ENGAGING STUDENTS • FOSTERING ACHIEVEMENT • CULTIVATING 21ST CENTURY
GLOBAL SKILLS**

Breaks between tasks
Cue expected behavior
Daily feedback to student
Have contingency plans
Use de-escalating strategies
Use positive reinforcement
Use proximity/touch control
Use peer supports and mentoring
Model expected behavior by adults
Have parent sign homework
Have parent sign behavior chart
Set and post class rules
Chart progress and maintain data



SPRINGFIELD PUBLIC SCHOOLS

**ENGAGING STUDENTS • FOSTERING ACHIEVEMENT • CULTIVATING 21ST CENTURY
GLOBAL SKILLS**

IEP AND 504 ACCOMODATIONS AND MODIFICATIONS FOR MIDDLE AND HIGH SCHOOL STUDENTS:

Large print textbooks
Textbooks for at-home use
Additional time for assignments
A locker with adapted lock
Review of directions
Review sessions
Use of mnemonics
Have student restate information
Provision of notes or outlines
Concrete examples
Adaptive writing utensils
Support auditory presentations with visuals
Weekly or bi-weekly home-school communication
Space for movement or breaks
Study sheets and teacher outlines
Extra visual and verbal cues and prompts
Lab and math sheets with highlighted instructions
Graph paper to assist in organizing or lining up math problems
Use of tape recorder for lectures
Use of Interactive notebook
Use of a scribe
Use of computers and calculators
Books on tape
Graphic organizers
Quiet corner or room to calm down and relax when anxious
Preferential seating
Alteration of the classroom arrangement
Reduction of distractions
Answers to be dictated
Hands-on activities
Ability to type all assignments
Follow a routine/schedule
Teach time management and organizational skills
Rest breaks
Verbal and visual cues regarding directions and staying on task
Agenda book and checklists
Daily check-in with case manager or special education teacher
Adjusted assignment timelines
Visual daily schedule



SPRINGFIELD PUBLIC SCHOOLS

**ENGAGING STUDENTS • FOSTERING ACHIEVEMENT • CULTIVATING 21ST CENTURY
GLOBAL SKILLS**

Varied reinforcement procedures
Immediate feedback
Work-in-progress check
Personalized examples

TESTING AND ASSESSMENT ACCOMMODATIONS AND MODIFICATIONS:

Answers to be dictated
Frequent rest breaks
Additional time
Oral testing
Choice of test format (multiple-choice, essay, true-false)
Alternate ways to evaluate (projects or oral presentations instead of written tests)
Accept short answers
Open-book or open-note tests
Read test and directions to student
Provide study guides prior to tests
Highlight key directions
Test in alternative site
Use of calculator
Extra credit option
Pace long-term projects
Preview test procedures
Simplified test wording; rephrased test questions and/or directions

GENERAL ACCOMMODATIONS AND MODIFICATIONS:

Allow outlining, instead of writing for an essay or major project
Use of alternative books or materials on the topic being studied
Computerized spell-check support
Word bank of choices for answers to test questions
Provision of calculator for math tests
Film or video supplements in place of reading text
Reworded questions in simpler language
Projects instead of written reports
Highlighting important words or phrases in reading assignments
Modified workload or length of assignments/tests
Modified time demands



SPRINGFIELD PUBLIC SCHOOLS

**ENGAGING STUDENTS • FOSTERING ACHIEVEMENT • CULTIVATING 21ST CENTURY
GLOBAL SKILLS**

Pass/no pass option

Modified grades based on IEP

BEHAVIOR ACCOMMODATIONS FOR STUDENTS:

- Breaks between tasks
- Cue expected behavior
- Daily feedback to student
- Have contingency plans
- Use de-escalating strategies
- Use positive reinforcement
- Use proximity/touch control
- Use peer supports and mentoring
- Model expected behavior by adults
- Have parent sign homework
- Set and post class rules
- Chart progress and maintain data



SPRINGFIELD PUBLIC SCHOOLS

ENGAGING STUDENTS • FOSTERING ACHIEVEMENT • CULTIVATING 21ST CENTURY
GLOBAL SKILLS

Accommodations and Modifications for ESL Students

- Give tests orally
- Give instructions/directions in writing and orally
- Assign a peer tutor, same language or English speaking
- Allow use of bilingual/translation dictionary
- Provide summaries of text or study guides of particular chapters
- Allow errors in speaking
- Allow errors in writing
- Accept writing in first language
- Highlight key vocabulary
- Reduce amount of work required
- Assess comprehension through demonstration or other alternative means (gestures, drawings)
- Rephrase questions, directions, and explanations
- Reduce multiple choices on assessments
- Provide study guides and/ or outlines
- Shorten reading assignments
- Allow extended time for test/project completion
- Use of audio books
- Use of graphic organizers
- Use reduced text, so that print is not dense
- Use visuals/pictures throughout lesson/unit/assignment
- Adapt homework to reflect language proficiency/home support
- Adapt homework to reflect home support
- Provide hands-on activities and explanations
- Allow extended time to answer questions, and permit drawing, as an explanation
- Use of translation dictionaries to locate words in the native language
- Use anecdotal records and portfolio of work as forms of assessment
- Use story retellings to assess comprehension