Unit 5: Advanced Topics: Dynamic Data Structures and Object-Oriented Programming

Content Area: **Technology**
Course(s): **Computer Programming C++ I & II**
Time Period: **1 marking period**
Length: **Weeks**
Status: **Published**

Unit Overview

In this unit students will learn what dynamic data structures are, what distinguishes dynamic data structures from ordinary static data structures, and how implementing dynamic data structures requires working with variables addresses and pointers. Students will also become acquainted with the concepts of object-oriented programming, including defining classes and constructing ojects, writing member functions, and understand the concept of polymorphism. Students will study recursion as one method of solving programming problems. Additionally, students will analyze and discuss the advantages and trade-offs of technology in society including its impact on personal privacy.

Transfer

At the completion of this unit students should be able to write programs using dynamic data structures (such as linked lists, queues, stacks, maps, and/or trees), and understand the types of problems that give rise to these data structures. Students should be able to explain the methodology of object-oriented programming and make use of pre-existing classes in the course of solving programming challenges. Students should be able to offer concrete reasons for their personal opinions and beliefs in regards to privacy issues in the digital age.

Meaning

Understandings

-Dynamic data structures, in contrast to static data structures, do not have their memory layout fixed at compile time and can grow or shrink in size

-Pointers can be used to access computer memory via its address

-The use of pointers together with dynamic memory allocation permit the construction of a wide variety of flexible dynamic data structures

-Object-oriented programming involves the concepts of objects, which tie together data and functions, and classes, that define the properties of objects

-Object-oriented programming methodology advocates data encapsulation, whereby the data in a class is not

directly accessible to external code, as well as polymorphism, whereby different classes of objects with different behavior share a common interface

-The issues of privacy and safety in the digital age are complex and involve trade-offs

-Corporations and governments have varying policies regarding privacy, which consumers and citizens need to be familiar with in order to make informed decisions

Essential Questions

-How do I declare, initialize, and dereference pointer variables?

-How is accessing memory with pointers different from use of memory through ordinary variables?

-Under what circumstances is it necessary or advantageous to access memory using pointers?

-What is the role of pointers in the construction of dynamic data structures such as linked lists?

-How does object-oriented programming simplify the construction of complex software systems?

-How is data encapsulation an advantage when it limits access to data?

-For what types of programming problems is it be advantageous to apply polymorphism, in which different classes of objects are accessed in a uniform way?

-What are some of the trade-offs in regards to privacy and security in the information age?

-What role should citizens play in determining the privacy and security policies of their governments or the companies they interact with?

Application of Knowledge and Skill

Students will know...

-Dynamic data structures include linked lists, vectors, stacks, queues, maps (associative arrays) and trees

-Dynamic memory allocation sets aside memory for data from an area of program memory known as the heap

-Dynamic memory needs to be manually allocated (with the keyword 'new'), and freed (with the keyword 'delete') to prevent memory leaks

-Dynamic data structures typically require dynamic memory allocation

-Pointer variables can store the address of another variable using the address-of operator, or can hold the

address of dynamically allocated memory

-Pointers allow memory to be accessed indirectly with the dereference operator

-Objects contain both data and functions and are constructed based on classes

-Classes are defined similarly to structs, but using the class keyword, and contain member functions in addition to data

-In classes, typcially, data is encapsulated (restricted from external access) using the private keyword

-Addressing the privacy issues of the digitial age (at personal, corporate and government levels) requires being informed of the facts

Students will be skilled at...

-Declaring and initializing pointer variables

-Accessing memory indirectly using pointers

-Allocating and freeing memory dynamically from the heap using the 'new' and 'delete' keywords

-Using dynamic data structures (e.g., linked lists or associative arrays)

-Using pre-existing classes to construct objects

-Defining simple classes containing private data and pulic member functions

-Providing reasons for their beliefs and opinions on important topics relating to privacy in the digital age

Academic Vocabulary

dynamic data structure, static data structure, pointer, address-of operator, dereference operator, the heap and dynamic memory allocation, the keywords 'new', 'delete' and 'class', linked list, associative array/map, graph, binary tree, object, class, recursion, the keywords 'private' and 'public', encapsulation, polymorphism, object-oriented programming, constructor, accessor and mutator functions

Learning Goal 1

Students will be able to use pointers to access memory, safely dynamically allocate memory from the heap, define dynamic data structures and compare and contrast dynamic data structures from ordinary static data structures. Additionally, students will investigate recursion as a method of solving programming problems

involving dynamic data structures.

Target 1

Students will be able to declare and initialize pointer variables and use them to access data in memory using the dereference and address-of operators and use models to illustrate the layout of variables with addresses in memory.

Additional Standards:

9.3.IT-PRG.1, 2, 9.3.ST.6

TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

Target 2

Students will be able to dynamically allocate memory from the heap, store the address in a pointer, access the memory using the pointer, and free the memory before the program terminates, diagnosing and correcting errors such as memory leaks.

Additional Standards:

9.3.ST-SM.1, 2

TECH.8.2.12.C.4	Explain and identify interdependent systems and their functions.
TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

Target 3

Students will be able to define dynamic data structures, and compare and contrast dynamic data structures with ordinary static data structures.

Additional Standards:

9.3.IT-PRG.10, 9.3.ST-SM.4

TECH.8.2.12.C.4	Explain and identify interdependent systems and their functions.
TECH.8.2.12.C.5	Create scaled engineering drawings of products both manually and digitally with materials and measurements labeled.

Target 4

Students will be able to use dynamic data structures in the course of solving programming problems.

Additional Standards:

9.3.IT-PRG.10, 9.3.ST-SM.4

TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

Target 5

Students will investigate recursion as a method of solving programming problems, e.g. computing Fibonnacci numbers or traversing a binary tree.

Additional Standards:

9.3.ST-SM.1, 2

TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

Learning Goal 2

Students will understand the concepts of object-oriented programming, including defining classes and constructing ojects, writing member functions, describe the reasons behind the concepts of encapsulation and polymorphism, and contrast object-oriented programming with structured programming.

Target 1

Students will explain the concepts of object-oriented programming methodology, including the role of classes, objects, encapsulation and polymorphism, and compare and contrast structured programming with object-oriented programming.

Target 2

Students will be able to construct and use objects from predefined classes (e.g., from the C++ Standard Library), and illustrate with diagrams the binding of data with functions underlying the concept of object in object-oriented programming.

Target 3

Students will be able to write simple classes with private (encapsulated) data and public constructor, accessor and mutator functions, and describe the reasons for doing so and the circumstances under which it provides conceptual simplification.

Learning Goal 3

Students will analyze advantages, disadvantages and trade-offs of technology, and take a position on issues pertaining to its impact on society and personal privacy. Topics of investigation include encryption technology, the vulnerability of ordinary unencrypted information exchange on the internet, and issues of national security.

Target 1

Students will be able to explain the function of data encryption and be able to explain its significance for financial transactions on the Internet.

Additional Standards:

9.3.ST-SM.1,2,4

TECH.8.2.12.D.6	Synthesize data, analyze trends and draw conclusions regarding the effect of a technology on the individual, society, or the environment and publish conclusions.

Target 2

Students will be able to write a simple substitution cipher encryption program and use it to encrypt and decrypt data, and investigate the properties of a lookup map that make the substitution reversible.

Additional Standards:

9.3.ST-SM.1,2,4

TECH.8.2.12.C.6	Research an existing product, reverse engineer and redesign it to improve form and function.
TECH.8.2.12.C.7	Use a design process to devise a technological product or system that addresses a global problem, provide research, identify trade-offs and constraints, and document the process through drawings that include data and materials.

Target 3

Students will analyze the advantages, disadvantages and trade-offs involved in different kinds of technology, and take a position on its relationship to personal privacy.

Additional Standards:

9.3.IT.4, 9.3.ST-SM.3, 9.3.ST-ET.2

TECH.8.2.12.D.4	Assess the impacts of emerging technologies on developing countries.
TECH.8.2.12.D.6	Synthesize data, analyze trends and draw conclusions regarding the effect of a technology on the individual, society, or the environment and publish conclusions.

Summative Assessment

Chapter and benchmark assessments

Project-based assessments, possily including:

-Slotmachine project

-Substitution cipher/Encryption project

-Space Shooter project

21st Century Life and Careers

CAEP.9.2.12.C.7	Examine the professional, legal, and ethical responsibilities for both employers and employees in the global workplace.
CAEP.9.2.12.C.8	Assess the impact of litigation and court decisions on employment laws and practices.

Formative Assessment and Performance Opportunities

Daily challenge tasks, individual and group project presentations, portfolio assessments, and in-class observation, demonstration and questioning.

Accommodations/Modifications

-Supplemental project guides for extra help

-One-on-one assistance

Enrichment:

-Bitshift operators

-Sophisticated algorithms such as the Fischer-Yates shuffle

-Extra challenge for projects:

-Optimizing the sustitution cipher

-Writing encrypted data to a file and decrypting it when reading the file

Unit Resources

-SDL Wrapper library for creating graphical programs

-Slotmachine project guides

-Substitution cipher/Encryption project guide

-The United States of Secrets PBS video series

-Space Shooter game guides

-www.cplusplus.com

Interdisciplinary Connections

MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
LA.WHST.11-12.4	Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
LA.WHST.11-12.6	Use technology, including the Internet, to produce, share, and update writing products in response to ongoing feedback, including new arguments or information.