Indiana Department of Education
Academic Standards Course Framework

COMPUTER SCIENCE II

Computer Science II: Programming explores and builds skills in programming and a basic understanding of the fundamentals of procedural program development using structured, modular concepts. Coursework emphasizes logical program design involving user-defined functions and standard structure elements. Discussions will include the role of data types, variables, structures, addressable memory locations, arrays and pointers and data file access methods. An emphasis on logical program design using a modular approach, which involves task oriented program functions.

- DOE Code: 5236
- Recommended Grade Level: Grade 10-12
- Required Prerequisite: Computer Science I
- Credits: 2 semester course, 2 semesters required, 1-3 credit per semester, maximum of 6 credits
- Qualifies as a quantitative reasoning course
- Counts as an elective or Directed Elective for all diplomas

Implementation Guidance

Computer Science II builds on skills developed in Introduction to Computer Science I. This course is designed to be taught in any programming language, meaning it is language neutral. Additionally, all domains are an intrinsic part of the software development lifecycle. All domains are expected to be covered by the end of the second semester.

Dual Credit

This course provides the opportunity for dual credit for students who meet postsecondary requirements for earning dual credit and successfully complete the dual credit requirements of this course. The Dual Credit crosswalk can be accessed here.

Career and Technical Student Organizations (CTSOs)

Career and Technical Student Organizations are considered a powerful instructional tool when integrated into Career and Technical Education programs. They enhance the knowledge and skills students learn in a course by allowing a student to participate in a unique program of career and leadership development. Students should be encouraged to participate in Business Professional of America, DECA, or Future Business Leaders of America, the CTSOs for this area.

Content Standards

Domain – Problem Analysis

Core Standard 1 Students analyze a problem using current industry techniques to design a solution to a problem.

Standards

CSII-1.1 Document problem analysis through industry standards such as: flowcharts, functional specifications, user stories, etc.

CSII-1.2 Outline the problem assigned and describe the solution.

CSII-1.3 Use puzzles and games to enhance problem solving skills.

CSII-1.4 Recognize language appropriate planning and designing tools (for example: flowcharts, UML diagrams, pseudocode, use cases).

CSII-1.5 Modify an existing program, such as a template, to add additional functionality and discuss intended and unintended implications.

CSII-1.6 Justify what programming methodology to use (object oriented or procedural).

Domain – Collaboration and Communication

Core Standard 2 Students collaborate to complete various tasks.

Standards

CSII-2.1 Design a solution to a problem by working in a team.

CSII-2.2 Explore technologies that can be used to collaborate with others of various cultures and career fields.

CSII-2.3 Utilize a problem solving approach to develop a solution using technology.

CSII-2.4 Analyze the work of peers and provide feedback.

CSII-2.5 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices in a group software project.

Domain – Algorithms, Data, and Analysis

Core Standard 3 Students design a solution to a problem using algorithms.

Standards

CSII-3.1 Develop algorithms to determine a solution.

CSII-3.2 Assess the use of algorithms to provide a solution.

CSII-3.3 Create pseudocode to describe a solution.

CSII-3.4 Create a program flowchart using ANSI standard flowcharting symbols to define a solution.

CSII-3.5 Explain how the algorithm can be used to solve a problem.

CSII-3.6 Apply Boolean logic and relational operations.

CSII-3.7 Demonstrate iteration in an algorithm.

CSII-3.8 Evaluate algorithms in terms of their efficiency, correctness, and clarity.

CSII-3.9 Illustrate the flow of execution of a recursive algorithm.

CSII-3.10 Integrate classic algorithms (sorting and searching) to solve computational problems.

CSII-3.11 Demonstrate the different methods for encoding data such as binary, decimal, hexadecimal, ASCII, and Unicode.

Domain – Program Development

Core Standard 4 Students create executable programs that utilize the Object Oriented Programming (OOP) or procedural paradigm. Such programs demonstrate use of control structures, modularity, scoping, variables, and at least one data structure such as a list/array.

CSII-4.1 Define the process of programming.

CSII-4.2 Plan and develop programs for broad audiences using a software development lifecycle process (such as: Agile, waterfall, and spiral).

CSII-4.3 Adhere to industry standard programming conventions for accuracy and readability.

CSII-4.4 Demonstrate code reuse by creating programming solutions using libraries and APIs.

CSII-4.5 Create an advanced computer program that corresponds to an algorithm or proposed solution.

CSII-4.6 Implement fundamental programming constructs, including data types, control structures, methods, and arrays.

CSII-4.7 Compare local scope and global scope.

CSII-4.8 Create programmer defined functions and methods to break down program logic and support reuse.

CSII-4.9 Recognize the order of operations used by a computer when performing calculations.

CSII-4.10 Implement simple and compound conditionals (Boolean - not, and, or).

CSII-4.11 Demonstrate the relationship between classes and objects (instances).

CSII-4.12 Demonstrate the differences between instance variables and class variables.

CSII-4.13 Demonstrate the differences between instance methods and class methods.

CSII-4.14 Apply inheritance, polymorphism, encapsulation, and abstraction in the context of Object Oriented Programming.

CSII-4.15 Compare the graphical user interface and the command line interface.

Domain – Program Verification and Debugging

Core Standard 5 Students prove a computer program solution works by using verification and debugging techniques.

Standards

CSII-5.1 Predict and explain programming outcomes.

CSII-5.2 Identify cause/effect for input/output.

CSII-5.3 Understand input validation.

CSII-5.4 Develop and use a series of test cases to verify that a program performs according to its design specifications.

CSII-5.5 Peer review documentation and code.

CSII-5.6 Differentiate between syntax, logic, and semantic errors.

CSII-5.7 Debug code using techniques such as: code tracing, print statements, boundary testing, breakpoints, etc.

CSII-5.8 Review computing artifacts to reduce bias, increase equity, and support accessibility.

Domain – Documentation

Core Standard 6 Students document their computing artifacts as part of the software development lifecycle.

Standards

CSII-6.1 Describe the function of a computing artifact (for example, code or design).

CSII-6.2 Identify the purposes of a computing artifact.

CSII-6.3 Explain concepts related to a computing artifact.

CSII-6.4 Describe how to use a computing artifact.

CSII-6.5 Explain cause/effect by interpreting input and output.

CSII-6.6 Create documentation for computing artifact such as comments or user manual/readme.

Domain – Security and Privacy

Core Standard 7 Students will be able to understand the laws and ethical implications of privacy, security, and intellectual property.

Standards

CSII-7.1 Examine the dynamic between privacy and security.

CSII-7.2 Explain the privacy concerns related to the collection and generation of data through implicit and explicit processes.

CSII-7.3 Evaluate the social and emotional implications of privacy in the context of safety, law, and ethics.

CSII-7.4 Give examples to illustrate how sensitive data can be affected by malware and other attacks.

CSII-7.5 Discuss the concepts and justifications for using secure design techniques.

CSII-7.6 Discuss the laws surrounding intellectual property.

Domain – Careers

Core Standard 8 Students will investigate various careers within the field of computer science.

Standards

CSII-8.1 Identify computer science occupations and the roles and responsibilities of each.

CSII-8.2 Report job outlook, demand, and projected wages for computer science careers.

CSII-8.3 Explore the job opportunities that are available in computer science.

CSII-8.4 Investigate post-secondary training opportunities and industry certifications that are available.