

DRAFT

COMMUNITY UNIT SCHOOL DISTRICT 200

Course **AP Computer Programming**
High School
Level **9 – 12**

1. Subject Expectation **Basic operations and concepts**
(NETS 1)

| | |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| Essential Learning 1 (Learning Standard) | Students demonstrate a sound understanding of the nature and operation of technology systems. |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------|

Critical Content

- State briefly a history of computers
- State the difference between the low-level and high-level programming languages
- Define various types of software and how they are used.
 - Language translators/compilers
 - Virtual machines
 - Operating systems
- Identify basic computer hardware and explain what it does
 - Primary and secondary memory
 - Processors
 - Peripherals
- Identify types of systems (i.e. single-user systems and networks)
- Describe the steps involved in program compilation and execution (compiler vs. interpreter)
- Investigate and know the binary number system
- List the basic stages involved in writing a computer application
- Discriminate the difference between top-down and object-oriented program design

- Identify the basic components and formal syntax of Java programs
 - Comments
 - Spelling
 - Syntax and semantics
 - Indentation
 - White space
 - Braces
 - Statements and semicolons
 - Class definition
 - Main method
 - Case-sensitive

| | |
|------------------------------------------------------|----------------------------------------------------------|
| Essential Learning 2 (Learning Standard) | Students are proficient in the use of technology. |
|------------------------------------------------------|----------------------------------------------------------|

Critical Content

- Know and use numeric types and expressions
 - Primitive data types and String
 - Declare and initialize variables and constants
 - Mathematical operators
 - Construct and evaluate expressions using precedence rules
 - Use and evaluate math methods in expressions
 - Assignment statements
- Use various input and output formats and methods
 - Various input / output methods
 - Escape sequences
 - Concatenation
 - Literal expressions
 - String operations
 - Formatting
- Use existing classes and objects
 - Import statements (libraries and classes)
 - Relationship between methods and objects (dot notation)
- Apply Standard Data Structures
 - Primitive data type (int, Boolean, double)
 - Classes
 - One-dimensional arrays
- Applets
 - Write an applet to perform a simple task
 - Construct a simple HTML Web page that executes an applet
- Graphics
 - Understand the coordinate system
 - Draw shapes including lines, rectangles and ovals

- Fill a shape
- Use colors
- Draw text
- Change foreground/background color
- Movement
- Apply Declarations
 - Constant declarations
 - Variable declarations
 - Class declarations
 - Interface declarations
 - Method declarations
 - Parameter declarations
- Differentiate between primitive data types and reference data types
- Standard Algorithms
 - Operations
 - Traversals
 - Insertions
 - Deletions
 - Searching
 - Sequential
 - Binary
 - Sorting
 - Selection
 - Insertion
 - Merge sort
- Decision structure
 - Implement selection control in a program by using if and if-else statements
 - Write Boolean expressions with relational and Boolean operators
 - Evaluate given Boolean expressions correctly.
 - Nest an if (if-else) statement inside another if (if-else) statement correctly
 - Implement switch statement
 - Understand break and switch statements
- Repetition statements
 - Construct various loops with the while statement
 - Construct various loops with the do-while statement
 - Construct various loops with the for statement
 - Understand and use recursion
 - Construct nested loops

2. Subject Expectation (NETS 2) Social, ethical, and human issues

| | |
|--------------------------------------------------|----------------------------------------------------------------------------------------------|
| Essential Learning 1 (Learning Standard) | Students understand the ethical, cultural, and societal issues related to technology. |
|--------------------------------------------------|----------------------------------------------------------------------------------------------|

Critical Content

- Responsible use of computer systems
 - System reliability
 - Privacy
 - Legal issues and intellectual property
 - Social and ethical ramifications of computer use
- Identify legal implications for misuse of software
- Identify the consequences surrounding inappropriate behavior and use of the Internet in students' school and district
- Reinforce the district's Authorization for Technology Access policy

3. Subject Expectation (NETS 6) (State Goal 8)

**Technology problem-solving and decision-making tools
Use algebraic and analytical methods to identify and describe patterns and relationships in data, solve problems and predict results.**

| | |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| Essential Learning 1 (Learning Standard) | Students employ technology for solving problems associated with systems of numbers using variables and patterns |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|

Critical Content

- 8.A.5
 - Apply appropriate control structures
 - Debugging
 - Categorize errors: compile-time, run-time, logic
 - Identify and correct errors
 - Techniques: use a debugger, add extra output statements, hand-trace code
 - Plan and write a program by applying learned concepts
 - Understand and modify existing code
 - Extend existing code using inheritance
- 8.B.5
 - Differentiate between applets and applications
 - Testing
 - Test classes and libraries in isolation
 - Identify boundary cases and generate appropriate test data
 - Perform integration testing

- 8.C.5
 - Identify and apply problem-solving techniques
 - Understand runtime exceptions
 - Identify pre- and post- conditions and assertions
 - Choose search or sort algorithm

**Essential Learning 2
(Learning Standard)**

Students employ technology for making informed decisions incorporating exponential and logarithmic functions to model situations

Critical Content

- Program Design
 - Read and understand a problem’s description, purpose and goals
 - Apply data abstraction and encapsulation
 - Read and understand class specifications and relationships among the classes
 - Understand and implement a given class hierarchy
 - Identify reusable components from existing code using classes and class libraries
- 8.C.5
 - Class Design
 - Design and implement a class
 - Design an interface
 - Choose appropriate data representation and algorithms
 - Apply functional decomposition
 - Extend a class using inheritance
 - Methodology
 - Object-oriented development
 - Top-down development
 - Encapsulation and information hiding
 - Procedural abstraction
 - Choose appropriate library classes and/or their methods
 - Compare running times of algorithms (i.e. sorts)
- 8.D.5
 - Numerical representations and limits
 - Representations of numbers in different bases
 - Limitations of finite representations (e.g. integer bounds, imprecision of floating-point representations, and round-off error)