**Curriculum**

**Advanced Placement Computer Science**

**Course Overview**

The course is intended to be equivalent to a first-year university course in Computer Science, and more specifically, to prepare students for the AP Exam in Computer Science. The course uses project-based assignments to introduce students to the concepts of object-oriented software design and programming, "elegant" coding style, documentation, implementation of algorithms using conditional statements, loops, and common searching and sorting algorithms. These topics are covered in the context of the Java, the programming language currently used on the AP test.

**Department Standards**

* Students will demonstrate creative thinking, construct knowledge, and develop innovative products and processes using technology.
* Students will use digital media and environments to communicate and work collaboratively, including at a distance, to support individual learning and contribute to the learning of others.
* Students will apply digital tools to gather, evaluate, and use information.
* Students will use critical thinking skills to plan and conduct research, manage projects, solve problems and make informed decisions using appropriate digital tools and resources.
* Students will understand human, cultural, and societal issues related to technology and practice legal and ethical behavior.
* Students will demonstrate a sound understanding of technology concepts, systems, and operations.

**Benchmarks**:

Design and implement computer-based solutions to problems in a variety of application areas.

Use and implement commonly-used algorithms and data structures.

Develop and select appropriate algorithms and data structures to solve problems.

Code fluently in an object-oriented paradigm using the programming language Java.

Read and understand a large program consisting of several classes and interacting objects.

Identify the major hardware and software components of a computer system, their relationship to one another, and the roles of these components within the system.

Recognize the ethical and social implications of computer use.

**Performance Indicators**

The ability of a student to read and understand problem descriptions, class specifications, and hierarchies.

The ability of a student to design and implement a class, choose appropriate data representation , select appropriate algorithms, and extend classes with inheritance.

The effectiveness with which a student implements programming constructs using object-oriented programming and procedural abstraction.

The ability of the student to understand, modify, and correct errors in programs they have written and programs written by others.

**Assessments**

QUARTER 1

10 Labs

9 Tests

QUARTER 2

6 Labs

4 Tests

2 Projects

QUARTER 3

9 Labs

3 Projects

4 Tests

QUARTER 4

GridWorld Exercises

3 Regular Tests

Mock AP Exam

Actual AP Test

Real-Time Music Applications

Building a Web Browser

**Core Topics**

Introduction to programming

Data Types

Arithmetic Expressions

Working with Strings

Converting Data Types

Pseudocode

Conditional Statements

Logical Operations

Arrays

Loops & Iteration

Static Methods

main Method

Overloading

Recursion

Classes & Instances

Objects

Compiling

The JVM

ArrayLists

Class Hierarchies

Polymorphism

Overriding Methods

Abstract Classes

Interfaces

Algorithms

Traversals, Replacements, Insertions & Deletions

Searching & Sorting

Intro to GridWorld

GridWorld Case Study:

GridWorld Classes

GridWorld Interfaces

Extending GridWorld Classes

AP Exam Review & Practice

Other (non-Java) programming languages

GUI-based programming

**Specific Content**

Variables: Integers, Doubles and Casting

Expressions: Arithmetic Expressions, Declaring and Assigning Values to Variables, Strings, Concatenation, String Methods, System.out Methods, Converting between Numbers and Strings, Booleans, Relations Operators, Comparing Strings, Logical Operators, Arrays

Program Control: Conditional Statements, Blocks, Iteration, While Loops, For Loops, For-Each Loops

Methods: Static Methods, main Method, Java Comments, Multiple Variable Declarations, Overloaded Methods, Recursive Methods

Object-Oriented Programming Concepts: Simple Objects, Classes and Instances, Public Classes, Arrays, ArrayLists

Inheritance and Polymorphism: Extending Classes, Class Hierarchies, Polymorphism, Overriding Methods

Class Definitions: Class Methods, Class Variables and Constants, final Block Variables, Multiple Constructors, Overloaded Instance Methods, Wrapper Classes, Access Modifiers, this, Object Aliasing

Abstractions: Abstract Classes, Interfaces, List<E> and Comparable<E> Interfaces

Algorithms: Traversals, Replacements, Insertions & Deletions

Searching and Sorting: Sequential Search, Binary Search, Selection Sort, Insertion Sort, Merge Sort

Program Analysis: Assertions and Exceptions

GridWorld Case Study Classes and Interfaces: Grid<E> Interface, Actors, Rocks, Flowers, Bugs, and Critters

Extending GridWorld Classes: Extending the Critter Class, Extending the Bug Class, Extending the Actor Class, Choosing a SuperClass to Extend

**Resources**