| **PRIMITIVE TYPES** | | | | |
| --- | --- | --- | --- | --- |
| Variables and Data Types  Expressions and Assignment  Statements  Compound Assignment Operators  Casting and Ranges of Variables  Introduction to an Integrated Development Environment  Introduction to an IDE | *Assignments may differ at teacher discretion*  **Formative**  Entry/Exit Slip Assessments  Participation  Discussions  [Short Programming Exercises](https://studycs.org/practice/view/1/) Quizzes CollegeBoard Unit 1 Progress Checks  **Summative**  Unit Test Project | Call System class methods to generate output to the console.  Create String literals.  Identify the most appropriate data type category for a articular specification.  Declare variables of the correct types to represent primitive data.  Evaluate arithmetic expressions in a program code.  Evaluate what is stored in a variable as a result of an expression with an assignment statement  Evaluate what is stored in a variable as a result of an expression with an assignment statement.  Evaluate arithmetic expressions that use casting.  Create java projects in an Integrated Development Environment (Eclipse, VS Studio)  **SKILLS** (Define at the bottom of the document): 1.A, 1.B, 2.B, 4.B, 5.A | C4.1  C4.3  C4.4  C4.6 | LS (11-12.1 to 11-12.6)  RSIT (11-12.1 to 12.7)  WS (11 - 12.1 to 12.6, 12.8, 12.9)  A-CED (1-4)  A-REI (1-2) F-IF (1-10)  F-LE (5-6)  C (5)  G-CO (6-8, 12) G-GMD (1-3, 5)  G-GPE (4-7)  G-MG (1, 3) G-SRT (1-3) N-RN (1-2)  N-Q (1-3) N-CN (4-6) N-VM (6-12) S-IC(1-6)  S-ID(1-6) S-MD (1-7)  APPS (10, 15, 16) |
| **USING OBJECTS** | | | | |
| Objects: Instances of Classes  Creating and Storing Objects  Calling methods  String Objects and Literals  String Methods  Wrapper Classes: Integer, Double  Using the Math Class | *Assignments may differ at teacher discretion*  **Formative**  Entry/Exit Slip Assessments  Participation  Discussions  Programming Exercises [Drawing Lab](https://studycs.org/project/view/drawing)  Quizzes  CollegeBoard Unit 2 Progress Checks  **Summative**  Unit Test | Explain the relationship between a class and an object.  Identify, using its signature, the correct constructor being called.   Create objects by calling different constructors  Define variables of the correct types to represent reference data.  Call non-static void methods with or without parameters.  Create String objects and use String class methods  Be familiar with the use of the Integer and Double wrapper classes.  Evaluate expressions that use the Math class methods.  Evaluate Boolean expressions that use relational operators in program code.  **SKILLS**:  1.B, 1.C, 2.A, 2.C, 3.A, 5.A | C4.7  C4.8 | LS (11-12.1 to 11-12.6)  RSIT (11-12.1 to 12.7)  WS (11 - 12.1 to 12.6, 12.8, 12.9)  A-CED (1-4)  A-REI (1-2) F-IF (1-10)  F-LE (5-6)  C (5)  G-CO (6-8, 12) G-GMD (1-3, 5)  G-GPE (4-7)  G-MG (1, 3) G-SRT (1-3) N-RN (1-2)  N-Q (1-3) N-CN (4-6) N-VM (6-12) S-IC(1-6)  S-ID(1-6) S-MD (1-7)  APPS (10, 15, 16) |
| **BOOLEAN EXPRESSIONS AND IF STATEMENTS** | | | | |
| Boolean Expressions  if Statements and Control Flow  if-else Statements  else if Statements  Compound Boolean Expressions  Equivalent Boolean Expressions  Comparing Objects  Introduction to Graphic-User-Interface in java | *Assignments may differ at teacher discretion*  **Formative**  Short Debugging Exercises [Programming Exercises](https://studycs.org/practice/view/3/)  [Introduction to GitHub](https://studycs.org/github/#introduction/) for Eclipse  Quizzes  CollegeBoard Unit 3 Progress Checks  **Summative**  [Pong](https://studycs.org/project/view/pong) - First GUI-Based Program  Unit Test | Represent branching logical processes by using conditional statements.  Compare and contrast equivalent Boolean expressions.  Compare object references using Boolean expressions in program code.  Follow starter code for a GUI-based java project  Write if-statements for keyboard input of a java program  Use java drawing methods to draw shapes in java  Complete a java program that uses a GUI for the simple game of Pong  **SKILLS**:  2.A, 2.B, 3.A, 3.C, 4.A, 4.C | C4.9 | LS (11-12.1 to 11-12.6)  RSIT (11-12.1 to 12.7)  WS (11 - 12.1 to 12.6, 12.8, 12.9)  A-CED (1-4)  A-REI (1-2) F-IF (1-10)  F-LE (5-6)  C (5)  G-CO (6-8, 12) G-GMD (1-3, 5)  G-GPE (4-7)  G-MG (1, 3) G-SRT (1-3) N-RN (1-2)  N-Q (1-3) N-CN (4-6) N-VM (6-12) S-IC(1-6)  S-ID(1-6) S-MD (1-7)  APPS (10, 15, 16) |
| **ITERATION** | | | | |
| while Loops  for Loops  Developing Algorithms for Strings  Nested Iteration  Code Analysis | *Assignments may differ at teacher discretion*  **Formative**  Entry/Exit Slip Assessments  Participation  Discussions Short Debugging Exercises [Programming Exercises](https://studycs.org/practice/view/4/)  Quizzes  CollegeBoard Unit 4 Progress Checks  **Summative**  Unit Test | Represent iterative processes using a while or for loop.  Identify, modify, and develop standard String-related algorithms  Represent nested iterative processes  Compute statement execution counts and informal run-time comparison of iterative statements.  **SKILLS**:  1.B, 2.B, 2.D, 3.C, 4.C, 5.C | C4.9 | LS (11-12.1 to 11-12.6)  RSIT (11-12.1 to 12.7)  WS (11 - 12.1 to 12.6, 12.8, 12.9)  A-CED (1-4)  A-REI (1-2) F-IF (1-10)  F-LE (5-6)  C (5)  G-CO (6-8, 12) G-GMD (1-3, 5)  G-GPE (4-7)  G-MG (1, 3) G-SRT (1-3) N-RN (1-2)  N-Q (1-3) N-CN (4-6) N-VM (6-12) S-IC(1-6)  S-ID(1-6) S-MD (1-7)  APPS (10, 15, 16) |
| **WRITING CLASSES** | | | | |
| Anatomy of a Class  Constructors  Code Comments  Accessor/Getter Methods  Mutator/Setter Methods  Writing Methods  Static Variables and Methods  Scope and Access  this Keyword | *Assignments may differ at teacher discretion*  **Formative**  Entry/Exit Slip Assessments  Participation  Discussions  Short Debugging Exercises [Programming Exercises](https://studycs.org/practice/view/5/) Quizzes  [Introduction to AP-Level FRQs Style 1 - Methods and Control Structures](https://studycs.org/frq/index/1/) [Introduction to AP-Level FRQs Style 2 - Classes](https://studycs.org/frq/index/2/)  CollegeBoard Unit 5 Progress Checks  **Summative**  Unit Test  [DuckHunt Project](https://docs.google.com/document/d/14PJhVcIOA3CPJ_rz8lsqJnmYFeGqBPoHOiGNe_0RZQ4/edit?usp=sharing) | Designate access and visibility constraints to classes, data, constructors, and methods.  Designate private visibility of instance variables to encapsulate the attributes of an object.  Define instance variables for the attributes to be initialized through the constructors of a class.  Describe the functionality and use of program code through comments.  Define behaviors of an object through non-void methods without parameters written in a class.  Define behaviors of a class through static methods.  Define the static variables that belong to the class.  Explain where variables can be used in the program code.  Evaluate object reference expressions that use the keyword this  **SKILLS**:  1.A,1.B, 1.C, 2.C, 3.B, 4.B, 5.A, 5.B, 5.D | C4.5  C4.8  C4.11 | LS (11-12.1 to 11-12.6)  RSIT (11-12.1 to 12.7)  WS (11 - 12.1 to 12.6, 12.8, 12.9)  A-CED (1-4)  A-REI (1-2) F-IF (1-10)  F-LE (5-6)  C (5)  G-CO (6-8, 12) G-GMD (1-3, 5)  G-GPE (4-7)  G-MG (1, 3) G-SRT (1-3) N-RN (1-2)  N-Q (1-3) N-CN (4-6) N-VM (6-12) S-IC(1-6)  S-ID(1-6) S-MD (1-7)  APPS (10, 15, 16) |
| **ARRAY** | | | | |
| Creation and Access  Traversing Arrays  Enhanced for Loops for Arrays  Reading, Modifying, and Developing Algorithms Using Arrays | *Assignments may differ at teacher discretion*  **Formative**  Entry/Exit Slip Assessments  Participation  Discussions  [Debugging and Programming Exercises](https://studycs.org/practice/view/6/)  Quizzes  CollegeBoard Unit 6 Progress Checks  **Summative**  Frogger / [Agar.io](https://docs.google.com/document/d/1peYeduB2PbgXWuLwSjA41v9z3ehOfao436V05cVj2Uc/edit?usp=sharing) Project  Unit Test | Represent collections of related primitive or object reference data using one dimensional (1D) array objects.  Traverse the elements in a 1D array using a while, for, or enhanced for loop.  Identify, modify and develop array traversals algorithms.  Integrate 1D Arrays, Loops, and Objects into a GUI-Based java program.  **SKILLS**:  1.C, 2.B, 4.B, 4.C, 5.D | C4.7 | LS (11-12.1 to 11-12.6)  RSIT (11-12.1 to 12.7)  WS (11 - 12.1 to 12.6, 12.8, 12.9)  A-CED (1-4)  A-REI (1-2) F-IF (1-10)  F-LE (5-6)  C (5)  G-CO (6-8, 12) G-GMD (1-3, 5)  G-GPE (4-7)  G-MG (1, 3) G-SRT (1-3) N-RN (1-2)  N-Q (1-3) N-CN (4-6) N-VM (6-12) S-IC(1-6)  S-ID(1-6) S-MD (1-7)  APPS (10, 15, 16) |
| **ARRAYLIST** | | | | |
| Introduction to ArrayList  ArrayList Methods  Traversing ArrayLists  Reading, Modifying and Developing Algorithms Using ArrayList  Searching - Sequential  Sorting: Insertion, Selection | *Assignments may differ at teacher discretion*  **Formative**  Entry/Exit Slip Assessments  Participation  Discussions [Debugging & Programming Exercises](https://studycs.org/practice/view/7/)  [FRQ Practice](https://studycs.org/frq/index/3/) (AP Prep)  Quizzes  Infection Simulation / Agar.io Clone Re-visited with ArrayList and dynamically adding objects into simulation  CollegeBoard Unit 7 Progress Checks  **Summative**  Unit Test | Represent collections of related object reference data using ArrayList objects.  Traverse ArrayList objects using while, for, and enhanced loops.  Identify, modify and develop algorithms involving traversals  Apply sequential/linear search algorithms to search for specific information in an array or ArrayList of  objects.  Apply selection sort and insertion sort algorithms to sort the elements of an array or ArrayList objects.  Compute statement execution counts and informal run-time comparison of sorting algorithms.  **SKILLS**:  1.B, 2.C, 2.D, 5.C | C4.7  C4.10 | LS (11-12.1 to 11-12.6)  RSIT (11-12.1 to 12.7)  WS (11 - 12.1 to 12.6, 12.8, 12.9)  A-CED (1-4)  A-REI (1-2) F-IF (1-10)  F-LE (5-6)  C (5)  G-CO (6-8, 12) G-GMD (1-3, 5)  G-GPE (4-7)  G-MG (1, 3) G-SRT (1-3) N-RN (1-2)  N-Q (1-3) N-CN (4-6) N-VM (6-12) S-IC(1-6)  S-ID(1-6) S-MD (1-7)  APPS (10, 15, 16) |
| **2D ARRAY** | | | | |
| Creation and Access  Traversing 2D Arrays  Reading, Modifying and Developing Algorithms using 2D Arrays | *Assignments may differ at teacher discretion*  **Formative**  Entry/Exit Slip Assessments  Participation  Discussions[Debugging & Programming Exercises](https://studycs.org/practice/view/8/)  [MagicSquares](https://docs.google.com/document/d/12NTe4iBj5ynmOSqusvgQ47H5vafA_wJrEL8B0prdrSA/edit?usp=sharing)  [FRQ Practice](https://studycs.org/frq/index/4/) (AP Prep)  Quizzes  CollegeBoard Unit 8 Progress Checks  [Picture Lab](https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-compscia-picture-lab-student-guide.pdf) / [2048](https://docs.google.com/document/d/1ajDy8JXvY-S79Wgk2Ic-YRQ3gj1nkJX5nMyD1gYfGXQ/edit?usp=sharing)  **Summative**  Unit Test | Represent collections of related primitive or object reference data using two-dimensional (2D) array objects.  Traverse 2D array objects using while, for, or enhanced for loops.  Identify or modify algorithms requiring 2D array traversals  Write program code to create, traverse, and manipulate elements in 2D array objects.  **SKILLS**:  1.B, 1.C, 2.B, 2.D, 4.A, | C4.7 | LS (11-12.1 to 11-12.6)  RSIT (11-12.1 to 12.7)  WS (11 - 12.1 to 12.6, 12.8, 12.9)  A-CED (1-4)  A-REI (1-2) F-IF (1-10)  F-LE (5-6)  C (5)  G-CO (6-8, 12) G-GMD (1-3, 5)  G-GPE (4-7)  G-MG (1, 3) G-SRT (1-3) N-RN (1-2)  N-Q (1-3) N-CN (4-6) N-VM (6-12) S-IC(1-6)  S-ID(1-6) S-MD (1-7)  APPS (10, 15, 16) |
| **INHERITANCE** | | | | |
| Creating Superclasses and Subclasses  Writing Constructors for Subclasses | **Formative**  Entry/Exit Slip Assessments  Participation  Discussions  [Programming Exercises](https://studycs.org/practice/view/9/)  [Elevens Lab](https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap-compscia-elevens-lab-student-guide.pdf)  Quizzes  CollegeBoard Unit 9 Progress Checks  **Summative**  Unit Test | Create an inheritance relationship from a subclass to the superclass  Define reference variables of a superclass to be assigned to an object of a subclass in the same hierarchy.  Call methods in an inheritance relationship.  Call Object class methods through inheritance.  **SKILLS:**  1.A, 1.C, 3.A, 3.B, 5.A, 5.D | C4.5  C4.8 | LS (11-12.1 to 11-12.6)  RSIT (11-12.1 to 12.7)  WS (11 - 12.1 to 12.6, 12.8, 12.9)  A-CED (1-4)  A-REI (1-2) F-IF (1-10)  F-LE (5-6)  C (5)  G-CO (6-8, 12) G-GMD (1-3, 5)  G-GPE (4-7)  G-MG (1, 3) G-SRT (1-3) N-RN (1-2)  N-Q (1-3) N-CN (4-6) N-VM (6-12) S-IC(1-6)  S-ID(1-6) S-MD (1-7)  APPS (10, 15, 16) |
| **RECURSION** | | | | |
| Determining the result of recursive  methods  Writing simple recursive methods:  Determining base cases and recursive  calls  Recursive Traversals on ArrayLists  and Strings  Recursive Searching Algorithm: Binary Search  Recursive Sorting Algorithm: Merge Sort | *Assignments may differ at teacher discretion*  **Formative**  [Programming Exercises](https://studycs.org/practice/view/10/)  [Fractal Drawing in Java - Recursvely Drawing Shapes](https://studycs.org/project/view/recursive)  Recursive Method Tracing Exercises  Quizzes  CollegeBoard Unit 10 Progress Checks  **Summative**  Unit Test Group Project | Apply recursive search algorithms to information in String, 1D array, or ArrayList objects.  Apply recursive algorithms to sort elements of an array or ArrayList objects.  Determine the result of executing recursive methods.  **SKILLS**:  1.B, 2.B, 2.D, 5.A, | C4.9  C4.10  C5.3  C5.6 | LS (11-12.1 to 11-12.6)  RSIT (11-12.1 to 12.7)  WS (11 - 12.1 to 12.6, 12.8, 12.9)  A-CED (1-4)  A-REI (1-2) F-IF (1-10)  F-LE (5-6)  C (5)  G-CO (6-8, 12) G-GMD (1-3, 5)  G-GPE (4-7)  G-MG (1, 3) G-SRT (1-3) N-RN (1-2)  N-Q (1-3) N-CN (4-6) N-VM (6-12) S-IC(1-6) S-MD (1-7)  APPS (10, 15, 16) |
| **ETHICAL COMPUTING** | | | | |
| Ethical and Social Implications of  Computing Systems  Ethical Issues Around Data Collection | **Formative**  Discussions | Explain the risks to privacy from collecting and storing personal data on computer systems. | C8.3  C8.6  C8.8  C10.1 | LS (11-12.1 to 11-12.6)  A-CED (1-4)  G-GPE (4-7)  G-MG (1, 3)  N-Q (1-3) N-CN (4-6) S-IC(1-6) S-MD (1-7) |

**SKILLS**

| 1.A  Determine an appropriate program design to solve a problem or accomplish a task.  1.B  Determine code that would be used to complete code segments.  1.C  Determine code that would be used to interact with completed program code. | 2.A  Apply the meaning of specific operators.  2.B  Determine the result or output based on statement execution order in a code segment without method calls (other than output).  2.C  Determine the result or output based on the statement execution order in a code segment containing method calls.  2.D  Determine the number of times a code segment will execute. | 3.A  Write program code to create objects of a class and call methods.  3.B  Write program code to define a new type by creating a class.  3.C  Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.  3.D  Write program code to create, traverse, and manipulate elements in a 1D array or ArrayList objects.  3.E  Write program code to create, traverse, and manipulate elements in 2D array objects. | 4.A Use test cases to find errors or validate results.  4.B  Identify errors in program code.  4.C  Determine if two or more code segments yield equivalent results | 5.A  Describe the behavior of a given segment of program code.  5.B  Explain why a code segment will not compile or work as intended.  5.C  Explain how the result of program code changes, given a change to the initial code.  5.D  Describe the initial conditions that must be met for a program segment to work as intended or described. |
| --- | --- | --- | --- | --- |