

Summer 2026 | AP Computer Science A Summer Assignment

CodeHS Textbook Activity Guide

Name:

Purpose

AP Computer Science A requires you to read, write, trace, debug, and explain Java programs with precision. This summer activity guide uses the CodeHS AP Computer Science A Textbook as the main resource so you can begin building Java fluency before the school year starts.

What to Make	Who It Is For	How It Will Be Graded
Written answers, Java methods, Java classes, and a final CourseGradeAnalyzer program.	Your future AP CSA self. This work should help you enter the course ready to use Java.	You will be graded on accuracy, code quality, reasoning, completion, and evidence that the work is your own.

Required Online Resources

Main resource: CodeHS AP Computer Science A Textbook

https://codehs.com/textbook/apcsa_textbook/

You do not need to be enrolled in our CodeHS section yet to read the textbook. Read the assigned textbook unit before each assignment. When school begins, we will use the official CodeHS section for assignments, progress tracking, and grading.

Reference resource: AP CSA Java Quick Reference

<https://apcentral.collegeboard.org/media/pdf/ap-computer-science-a-java-quick-reference.pdf>

Use this reference when checking String, Math, and ArrayList methods. It is a reference sheet, not an answer key.

General Directions

- ▶ Open the assigned CodeHS textbook unit and read the listed sections before starting each assignment.
- ▶ Take brief notes in your own words before answering questions or writing code.
- ▶ Write complete sentences for written responses.
- ▶ Test Java code in an online Java compiler, CodeHS Java Sandbox, or another Java environment.
- ▶ Keep a resource log of any websites, videos, AI tools, notes, tutors, classmates, or family members that helped you.

- ▶ Submit your written responses and Java files during the first week of school.

Assignment Map

Assignment	CodeHS Unit	Main Skill	Points
1	Unit 1: Primitive Types	Java basics, variables, arithmetic, modulus	25
2	Unit 2: Using Objects	Strings, objects, method calls	25
3	Unit 3: Boolean Expressions and if Statements	Conditionals and Boolean logic	25
4	Unit 4: Iteration	Loops, tracing, debugging	30
5	Unit 5: Writing Classes	Classes, constructors, accessors, mutators	35
6	Unit 6: Arrays	Array traversal and algorithms	30
7	Unit 7: ArrayLists	ArrayList methods and traversal	25
8	Units 1 through 7	Final CourseGradeAnalyzer project	40
9	Your own project code	AP CSA style written reflection	15
10	Your work process	Academic integrity and resource log	10

Assignment 1: Java Basics and Primitive Types

CodeHS Textbook Resource: Unit 1: Primitive Types

25 Points

BEFORE YOU BEGIN: REVIEW THESE CONCEPTS IN THE CODEHS TEXTBOOK FIRST

- Variables and data types
- int, double, and boolean
- Arithmetic operators
- Integer division
- Modulus %
- Assignment statements

Part A: Vocabulary and Explanation

1. What is a variable?

2. What is the difference between int and double?

3. What is the difference between = and ==?

4. What does the modulus operator % do?

5. Why can integer division cause unexpected results in Java?

Part B: Code Tracing

Study the code below. Predict the exact output of all three print statements and explain your reasoning for each.

```
int a = 19;
int b = 4;

System.out.println(a / b);
System.out.println(a % b);
System.out.println((double) a / b);
```

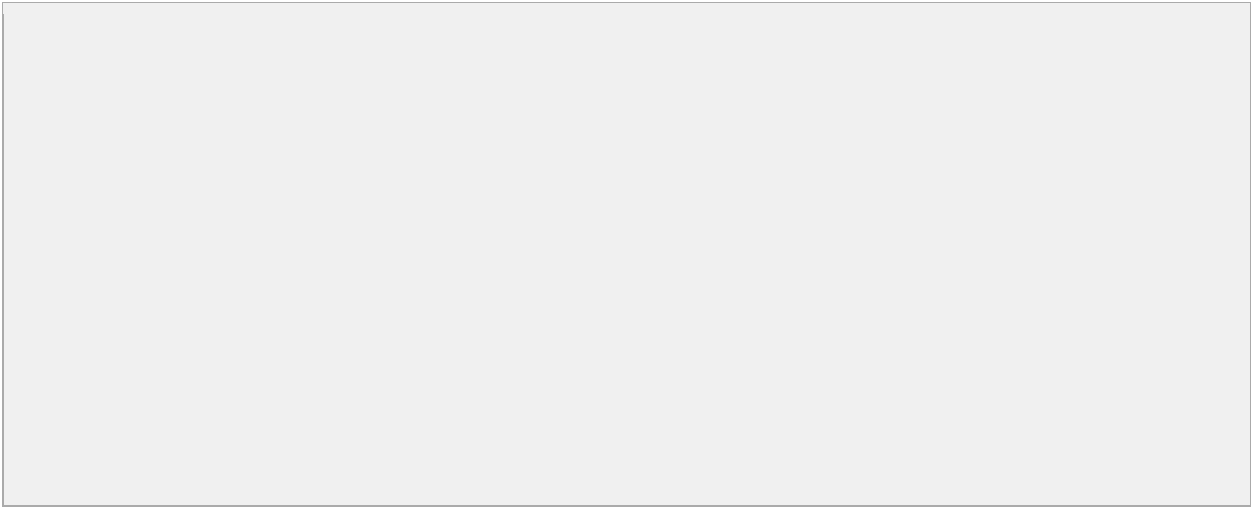
6. Predict the exact output of all three print statements. Explain your reasoning for each line.

Part C: Small Program

Write a Java program called SummerBudget that meets all of the following requirements.

- ▶ Store an amount of money earned over the summer.
- ▶ Store the cost of one item you want to buy.
- ▶ Calculate how many of that item you can afford.
- ▶ Calculate how much money will be left over.
- ▶ Print the results clearly.

7. Write the complete SummerBudget Java program below. You will also submit the .java file separately.



Assignment 2: Strings and Using Objects

CodeHS Textbook Resource: Unit 2: Using Objects

25 Points

BEFORE YOU BEGIN: REVIEW THESE CONCEPTS IN THE CODEHS TEXTBOOK FIRST

- Objects and object references
- String objects
- Calling methods
- length
- substring
- indexOf
- Return values

Part A: Code Tracing

Study the code below. Predict the exact output of each print statement and explain what each line does.

```
String word = "computer";

System.out.println(word.length());
System.out.println(word.substring(0, 4));
System.out.println(word.substring(3));
System.out.println(word.indexOf("put"));
System.out.println(word.indexOf("z"));
```

8. Predict the exact output of each print statement. Explain what each line does.

Part B: Written Questions

9. Why is a String considered an object instead of a primitive type?

10. What does it mean to call a method on an object?

11. What is the difference between `substring(0, 3)` and `substring(3)`?

12. Why do `String` index positions begin at 0?

Part C: Method Practice

Write the method described below. The method header is provided.

```
public static String makeUsername(String firstName, String lastName) {  
    // your code here  
}
```

Requirements: return the first three letters of the first name, the first three letters of the last name, and the combined length of both names concatenated together.

13. Write the complete body of the `makeUsername` method in the box below.

Assignment 3: Boolean Logic and Conditionals

CodeHS Textbook Resource: Unit 3: Boolean Expressions and if Statements

25 Points

BEFORE YOU BEGIN: REVIEW THESE CONCEPTS IN THE CODEHS TEXTBOOK FIRST

- Relational operators
- if, else if, and else
- &&, ||, and !
- Compound Boolean expressions
- Order of evaluation

Part A: Trace the Code

Study the code below, then answer the questions that follow.

```
int grade = 82;
boolean missingWork = true;
boolean extraCredit = false;

if (grade >= 90 && !missingWork) {
    System.out.println("Honor Roll");
} else if (grade >= 80 && !missingWork) {
    System.out.println("Strong Standing");
} else if (grade >= 70 || extraCredit) {
    System.out.println("Passing");
} else {
    System.out.println("Needs Support");
}
```

14. What is printed when this code runs with the values shown above?

15. Which condition is the first one that evaluates to true? Explain why.

16. Why does the second condition (`grade >= 80 && !missingWork`) fail? Be specific.

17. What would print if missingWork were false? Explain your reasoning.

18. What would print if grade were 65 and extraCredit were true? Explain which branch executes and why.

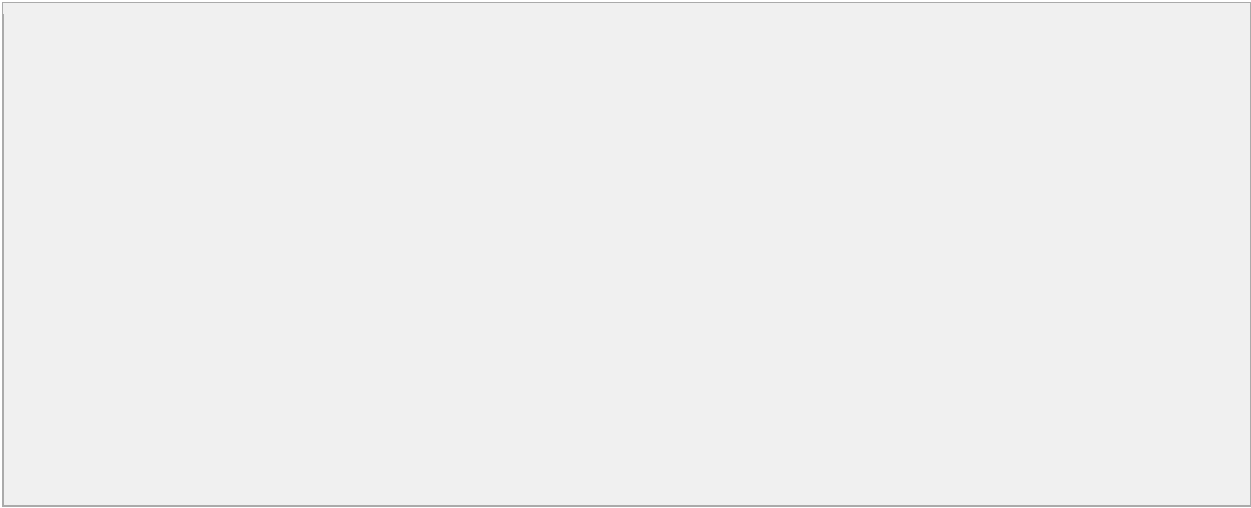
Part B: Write a Method

Write a method called getClubEligibility based on the requirements in the table below. You must use at least one compound Boolean expression.

Condition	Return Value
Grade is 90 or higher, no missing work, and no discipline referral	Gold Member
Grade is 75 or higher and no discipline referral	Eligible
Grade is 75 or higher but has a discipline referral	Conditional
Grade is below 75	Not Eligible

```
public static String getClubEligibility(int grade, boolean hasMissingWork, boolean
hasDisciplineReferral) {
    // your code here
}
```

19. Write the complete body of the getClubEligibility method in the box below.



Assignment 4: Iteration and Algorithm Tracing

CodeHS Textbook Resource: Unit 4: Iteration

30 Points

BEFORE YOU BEGIN: REVIEW THESE CONCEPTS IN THE CODEHS TEXTBOOK FIRST

- for loops
- while loops
- Loop counters
- Accumulators
- Counting patterns
- Off by one errors
- Using % inside loops

Part A: Trace the Loop

Study the code below, then answer the questions that follow.

```
int total = 0;
int count = 0;

for (int i = 1; i <= 20; i++) {
    if (i % 4 == 0 || i % 6 == 0) {
        total += i;
        count++;
    }
}

System.out.println(count);
System.out.println(total);
```

20. Which values of *i* are added to total? List them all.

21. What is printed by the two `System.out.println` statements? Show the values.

22. Why is `||` used instead of `&&` in the condition? What would happen if `&&` were used?

23. What would change about the output if the loop condition were `i < 20` instead of `i <= 20`?

Part B: Debug the Code

This method is supposed to count the number of even integers from 1 through `n`. It contains two logic errors.

```
public static int countEvens(int n) {  
    int count = 0;  
  
    for (int i = 1; i < n; i++) {  
        if (i / 2 == 0) {  
            count++;  
        }  
    }  
  
    return count;  
}
```

24. Identify the two logic errors in this method. Describe each error specifically.

25. Explain why each error causes an incorrect result. What does the method actually count instead?

26. Rewrite the corrected method in the box below.

Part C: Write a Method

Write the method described below. The method header is provided.

```
public static int sumMultiples(int limit, int factor) {  
    // your code here  
}
```

Requirements: return the sum of all positive integers from 1 through limit that are evenly divisible by factor.

27. Write the complete body of the sumMultiples method in the box below.

Assignment 5: Writing Classes

CodeHS Textbook Resource: Unit 5: Writing Classes

35 Points

BEFORE YOU BEGIN: REVIEW THESE CONCEPTS IN THE CODEHS TEXTBOOK FIRST

- Class headers
- Instance variables
- private visibility
- Constructors
- Accessor methods
- Mutator methods
- Object state
- Object behavior

Part A: Create a Class

Create a Java class called Book. Your class must include all of the following.

```
// Required private instance variables:  
private String title;  
private String author;  
private int pages;
```

- ▶ A constructor that initializes all three instance variables.
- ▶ Accessor methods for all three variables.
- ▶ Mutator methods for title and pages.
- ▶ A method called isLongBook that returns true if the book has more than 300 pages.
- ▶ A method called getReadingTime that returns the estimated number of minutes needed to read the book. Assume a reader reads 2 pages per minute.
- ▶ A method called getBookInfo that returns a sentence describing the book.

Example output from getBookInfo: The Hunger Games by Suzanne Collins has 374 pages.

28. Write the complete Book class in the box below. You will also submit the Book.java file separately.

Part B: Written Explanation

29. Why should instance variables usually be private?

30. What is the purpose of a constructor?

31. What is the difference between an accessor and a mutator?

32. Why is `getReadingTime` better as a method than as a separate stored variable?

33. How does the `Book` class model a real-world object?

Assignment 6: Arrays and Array Algorithms

CodeHS Textbook Resource: Unit 6: Arrays

30 Points

BEFORE YOU BEGIN: REVIEW THESE CONCEPTS IN THE CODEHS TEXTBOOK FIRST

- Creating arrays
- Setting elements
- Getting elements
- Indexing
- array.length
- Traversing arrays
- Enhanced for loops
- Developing algorithms with arrays

Part A: Trace the Code

Study the code below, then answer the questions that follow.

```
int[] scores = {84, 91, 73, 88, 95, 67};

int total = 0;
int count = 0;

for (int i = 0; i < scores.length; i++) {
    if (scores[i] >= 80) {
        total += scores[i];
        count++;
    }
}

System.out.println(total);
System.out.println(count);
System.out.println((double) total / count);
```

34. Which values are included in total? List them all.

35. What are the exact values printed by all three System.out.println statements?

36. Why is the cast (double) used in the final print statement?

37. What problem could occur if no scores were 80 or higher? What type of error would this cause?

Part B: Write Array Methods

Write all three methods below. Each method header is provided.

```
public static int findMax(int[] values)

public static int countAboveAverage(int[] values)

public static boolean containsValue(int[] values, int target)
```

38. Write the complete body of all three methods in the box below.

Part C: Algorithm Explanation

Choose one of the three methods you wrote in Part B.

39. Identify which method you chose. Explain what the loop does, what variable stores progress, what condition is checked, and why the method produces the correct result.

Assignment 7: ArrayLists

CodeHS Textbook Resource: Unit 7: ArrayLists

25 Points

BEFORE YOU BEGIN: REVIEW THESE CONCEPTS IN THE CODEHS TEXTBOOK FIRST

- Creating an ArrayList
- add
- remove
- get
- set
- size
- Index shifting
- Traversing an ArrayList

Part A: Trace the Code

Study the code below, then answer the questions that follow.

```
ArrayList<String> names = new ArrayList<String>();

names.add("Maya");
names.add("Jordan");
names.add("Ali");
names.add("Sofia");

names.remove(1);
names.add(1, "Diego");

System.out.println(names);
System.out.println(names.get(2));
System.out.println(names.size());
```

40. What is printed by all three System.out.println statements? Show exact output.

41. Which name was removed when names.remove(1) was called?

42. At what index was Diego inserted? What was at that index before the insertion?

43. What does `names.get(2)` return after all the modifications?

44. What does `names.size()` return? Explain why.

45. How is an `ArrayList` different from a standard array in Java? Name at least two differences.

Part B: Write an `ArrayList` Method

Write the method described below. The method header is provided.

```
public static int countLongNames(ArrayList<String> names) {  
    // your code here  
}
```

Requirements: return the number of names in the list that have more than five characters.

46. Write the complete body of the `countLongNames` method in the box below.

Part C: Short Explanation

47. Explain why removing an item from the middle of an ArrayList can affect the index positions of the remaining elements. Include an example.

Assignment 8: Final Summer Project: CourseGradeAnalyzer

40 Points

CodeHS Textbook Resource: Units 1 through 7

BEFORE YOU BEGIN: REVIEW THESE CONCEPTS IN THE CODEHS TEXTBOOK FIRST

- Primitive types
- Boolean expressions
- Iteration
- Writing classes
- Arrays
- ArrayLists
- Helper methods

Project Description

Create a Java program called CourseGradeAnalyzer. Your program must include two classes: Assignment and CourseGradeAnalyzer. Both classes are described below. You will submit the .java files during the first week of school.

Class 1: Assignment

Create a class called Assignment with the following private instance variables:

```
private String name;  
private int earnedPoints;  
private int possiblePoints;
```

- ▶ A constructor.
- ▶ Accessor methods for all instance variables.
- ▶ Mutator methods for earnedPoints and possiblePoints.
- ▶ A method called getPercentage that returns the assignment percentage as a double.
- ▶ A method called isMissing that returns true if possiblePoints is greater than 0 and earnedPoints is 0.
- ▶ A method called getSummary that returns a sentence like: Quiz 1: 18/20 = 90.0%

Class 2: CourseGradeAnalyzer

Create a class called CourseGradeAnalyzer. It must include a main method that does the following:

- ▶ Creates at least five Assignment objects.
- ▶ Stores the assignment objects in an array or ArrayList.
- ▶ Calculates total earned points.
- ▶ Calculates total possible points.
- ▶ Calculates the overall course percentage.

- Counts how many assignments are missing.
- Prints a summary of each assignment.
- Prints the overall course grade.
- Uses at least two helper methods besides main.

Required Helper Methods

Your program must include at least two of the following helper methods:

```
public static int getTotalEarned(ArrayList<Assignment> assignments)
public static int getTotalPossible(ArrayList<Assignment> assignments)
public static int countMissing(ArrayList<Assignment> assignments)
public static String getLetterGrade(double percentage)
```

Sample Output

```
Assignment Summary:
Quiz 1: 18/20 = 90.0%
Project 1: 42/50 = 84.0%
FRQ Practice: 0/15 = 0.0%
Homework 1: 10/10 = 100.0%
Test 1: 76/100 = 76.0%

Total Earned: 146
Total Possible: 195
Overall Percentage: 74.87%
Letter Grade: C
Missing Assignments: 1
```

Note: This assignment does not include written answer boxes. Submit your two .java files during the first week of school.

Assignment 9: AP CSA Style Written Reflection

CodeHS Textbook Resource: Your own project code

15 Points

Answer all questions below in complete sentences using your own CourseGradeAnalyzer program as your reference.

48. What is the purpose of your program?

49. What class did you create, and what real-world object or concept does it represent?

50. What method in your program uses a loop?

51. What does that loop do? Explain step by step.

52. How does one of your helper methods manage complexity? What would happen if that method did not exist?

53. What specific test case did you use to check that your program produces the correct output?

54. What logic error did you encounter or could you have encountered while writing this program?

55. How would you fix or prevent that error?

Assignment 10: Academic Integrity and Resource Log

10 Points

CodeHS Textbook Resource: Your work process

Answer honestly. Using resources to learn is allowed. Submitting work you do not understand is not allowed.

56. What CodeHS textbook sections did you use?

57. What other websites, videos, AI tools, notes, people, or resources helped you?

58. Which parts of the code did you write independently without any outside help?

59. Which concept was hardest for you to understand this summer?

60. What question do you still have before starting AP CSA in the fall?

61. Why is it important that your summer work reflects your own understanding?

Submission Checklist

- ▶ Written responses for Assignments 1 through 7 (this activity guide, fully completed).
- ▶ Java code files: SummerBudget.java, Book.java, Assignment.java, and CourseGradeAnalyzer.java.
- ▶ AP CSA style written reflection (Assignment 9).
- ▶ Academic integrity and resource log (Assignment 10).

200 Point Gradebook Rubric

Category	Points
Java basics, Strings, Boolean logic	50
Loops, tracing, and debugging	35
Classes and object-oriented programming	35
Arrays and ArrayLists	35
Final CourseGradeAnalyzer project	35
Written reflection and resource log	10
Total	200