

AP Computer Science A/P

Trial Lesson

Section	Question Type/Component	Number of Questions	Exam Weighting	Timing
I	Multiple-choice questions	70	70%	120 minutes End-of-course AP Exam
	Single-select	57		
	Single-select with reading passage about a computing innovation	5		
	Multi-select	8		
II	Create Performance Task	See Below	30%	See Below
	Program code, video, and Personalized Project Reference			At least 9 hours in class
	Written response questions related to the Create performance task	2		60 minutes End-of-course AP Exam

Big Ideas	Exam Weighting
Big Idea 1: Creative Development	10–13%
Big Idea 2: Data	17–22%
Big Idea 3: Algorithms and Programming	30–35%
Big Idea 4: Computer Systems and Networks	11–15%
Big Idea 5: Impact of Computing	21–26%

What Is AP Computer Science Principle?

- A broad introduction to computer science concepts
- Designed for beginners with no prior programming experience
- Focuses on creativity and problem-solving
- Explores how computing affects the world

What are we going to learn in this course?

- The basics of programming and algorithms
- Working with data
- The internet and cybersecurity
- Creative computing projects
- The impact of computing on society

Why should you learn AP CSP?

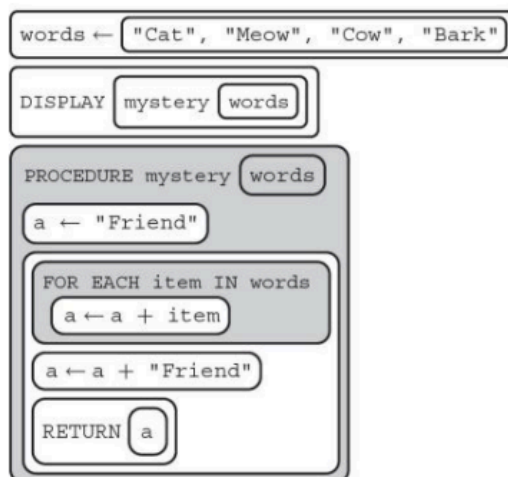
- You don't need coding experience — anyone can start!
- It helps you solve problems and think logically
- Computing is everywhere in daily life
- You'll build something real and creative
- It prepares you for any future — not just tech careers

1. Consider the following code segment, which uses the variables a and c .

```
 $a \leftarrow 3$   
 $a \leftarrow a + 5$   
 $c \leftarrow 3$   
 $a \leftarrow c + a$   
  
DISPLAY( $a$ )  
DISPLAY( $c$ )
```

3. An image stored on a computer contains pixels that represent how bright the red, green, and blue values are. The most common format for pixels is to represent the red, green, and blue, using 8 bits, which vary from 0 to 255. If the current red value is 10011101, what would be the new value in binary if the red value is increased by 4 in decimal?

- (A) 157_{BIN}
- (B) 0100_{BIN}
- (C) 10011111_{BIN}
- (D) 10100001_{BIN}



- (A) Friend
- (B) FriendFriend
- (C) FriendCatMeowCowBarkFriend
- (D) a

```

def check_answer(user_guess):
    # The 'global' keyword is used to modify variables defined outside the procedure
    global score
    global current_question_index

    # --- SEQUENCING ---
    # Step 1: Get the correct country/capital pair from the main list
    correct_pair = quiz_data[current_question_index]
    # Step 2: Extract the correct capital city from the pair
    correct_capital = correct_pair[1]

    # --- SELECTION (if/else) ---
    # Step 3: Compare the user's guess to the correct answer.
    # The .lower() method is used to make the comparison case-insensitive.
    if user_guess.lower() == correct_capital.lower():
        # If correct, increase the score
        score = score + 1
        print("Correct!")
    else:
        # If incorrect, show the correct answer
        print(f"Incorrect. The correct answer is {correct_capital}.")

    # --- ITERATION (Implicitly part of the game loop) ---
    # Step 4: Move to the next question for the next round
    current_question_index = current_question_index + 1

# --- A simple loop to run the quiz ---
# This loop demonstrates the iteration that calls the procedure repeatedly.
while current_question_index < len(quiz_data):
    # Get the current country
    current_country = quiz_data[current_question_index][0]

    # Ask the user for input
    guess = input(f"What is the capital of {current_country}? ")

    # Call the procedure to check the answer
    check_answer(guess)

    # Print current score
    print(f"Your score is: {score}")
    print("-----")

print("Quiz finished!")
print(f"Your final score is {score} out of {len(quiz_data)}.")

```

2. Refer to your Personalized Project Reference when answering this question.
 - (a) Consider the first iteration statement included in the Procedure section of your Personalized Project Reference. Describe what is being accomplished by the code in the body of the iteration statement.
 - (b) Consider the procedure identified in part (i) of the Procedure section of your Personalized Project Reference. Write two calls to your procedure that each cause a different code segment in the procedure to execute. Describe the expected behavior of each call. If it is not possible for two calls to your procedure to cause different code segments to execute, explain why this is the case for your procedure.
 - (c) Suppose another programmer provides you with a procedure called `checkValidity(value)` that returns `true` if a value passed as an argument is considered valid by the other programmer and returns `false` otherwise. Using the list identified in the List section of your Personalized Project Reference, explain in detailed steps an algorithm that uses `checkValidity` to check whether all elements in your list are considered valid by the other programmer. Your explanation must be detailed enough for someone else to write the program code for the algorithm that uses `checkValidity`.