What is the AP CSP?

The Advanced Placement Computer Science Principles (AP CSP) is a college-level introductory course that explores how computers, software, data, and the internet work and how they impact the world around us.

What you will learn:

- Design and evaluate programs to solve real-world problems
- Understand how data is stored, represented, and used
- Explain how computing systems and networks (like the internet) operate
- Analyze the social, ethical, and economic impacts of computing

Exam Format

There are two components of the Exam.

Performance Task

Students will **design and build a computer program that solves a real-world problem.**

End of Course Exam

A 120 minute, **70-question multiple-choice exam** that measures students computational thinking across the **five Big Ideas** of AP CSP.

70%
End of Course Exam

30%
Performance Task

5 Big Ideas of AP CSP

Creative Development	Algorithms and Programming	Data	Computer System Networks	Impact of Computing
Students learn how computer programs and digital products are designed, built, tested, and improved. This includes brainstorming ideas, building prototypes, debugging, and refining projects based on feedback.	Students learn how to write and understand step-by-step instructions (algorithms) and turn them into programs using variables, loops, conditionals, and lists.	Students learn how data is collected, stored, analyzed, and used to make decisions. They also explore how data can be misleading, biased, or used unethically.	Students learn how data is collected, stored, analyzed, and used to make decisions. They also explore how data can be misleading, biased, or used unethically.	Students learn how data is collected, stored, analyzed, and used to make decisions. They also explore how data can be misleading, biased, or used unethically.

AP Programming Reference Sheet

Assignment, Display, and Input

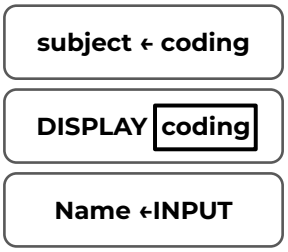
Coding Snippet

```
subject = "coding"

print(subject)

Name = input()
```

Text Snippet

```
subject ← coding

DISPLAY(subject)

Name ←INPUT()
```

Block Snippet



Mathematical Operations

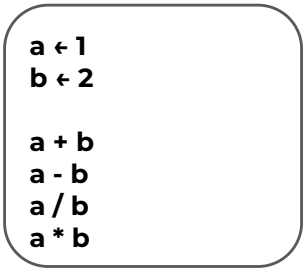
Coding Snippet

```
a = 1
b = 2

a + b
a - b
a / b
a * b
```

Text Snippet

```
a ← 1
b ← 2

a + b
a - b
a / b
a * b
```

Block Snippet



Procedures = Functions

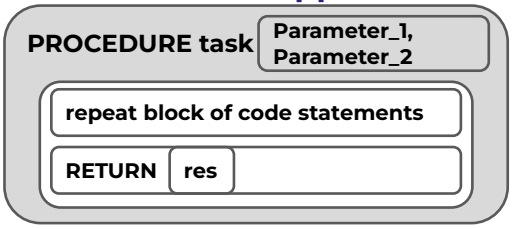
Coding Snippet

```
# function definition
def task( parameter1, parameter2 ) :
        # < block of code statements >
        res = result
        return res

# function invocation
task(1, 2)
```

Text Snippet

```
PROCEDURE task(parameter_1, parameter_2)
{
< block of code statements >
RETURN res
}
```

Block Snippet



AP Programming Reference Sheet

Relational and Boolean Operators

Coding Snippet

```
a == b
b != b

a > b
a < b
a < b
a >= b
a <= b
```

Text Snippet

```
a == 1
b ≠ 2

a > b
a < b
a ≥ b
a ≤ b
```

Block Snippet

```
a == 1
b ≠ 2

a > b
a < b
a ≥ b
a ≤ b
```

Written the same way

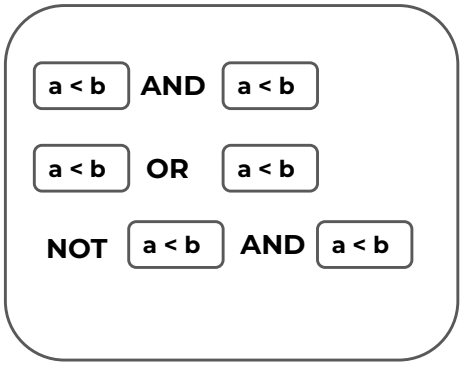
Coding Snippet

```
a = 1
b = 2

# Conditions
a < b and a > b

a > b or a < b

not(a > b and b > a)
```

Text Snippet

```
a = 1
b = 2

a < b AND  a > b

a < b OR  a > b

NOT a < b AND  a > b
```

Block Snippet



Lists

Coding Snippet

```
shopping_cart =
["coding",  1,  2, True]
```

Text Snippet

```
shopping_cart ← [coding, 1, 2, True]
```

Block Snippet

List Method

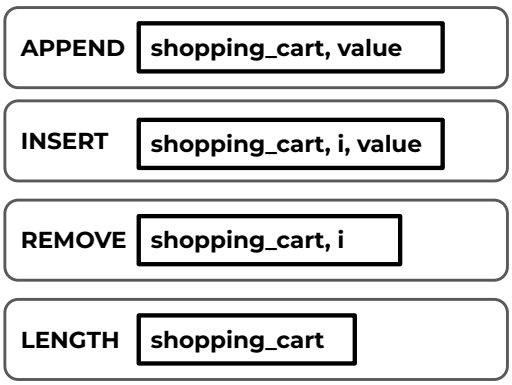
Coding Snippet

```
shopping_cart = ["coding",  1,  2,
True]

shopping_cart.append(value )
shopping_cart.insert(3, value)
shopping_cart.remove(1)

number_of_items =
len(shopping_cart)
```

Text Snippet

```
shopping_cart ← [coding, 1, 2, True]


APPEND(shopping_cart, value)
INSERT(shopping_cart, i, value)
REMOVE(shopping_cart, i)

LENGTH(shopping_cart)
```

Block Snippet



AP Programming Reference Sheet

Selection = Conditionals

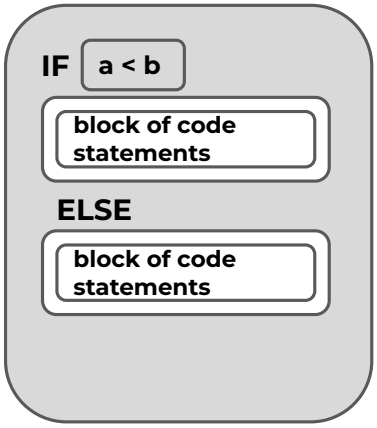
Coding Snippet

```
a = 1
b = 2

if a > b :
        # < block of
        code
        statements>
```

Text Snippet

```
a ← 1
b ← 2

IF(a > b)
{
<block of code
statements>
}
ELSE
{
<block of code
statements>
}
```

Block Snippet



Iteration = For Loops

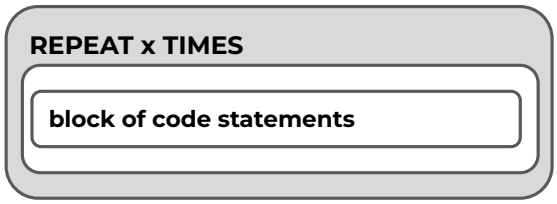
Coding Snippet

```
shopping_cart =
["coding", 1, 2, True]

for x in shopping_cart :
        # < repeat block of
        code statements
        x number of times
        >
```

Text Snippet

```
REPEAT x TIMES
{
<repeat block of code statements
x number of times >
}
```

Block Snippet



Iteration = While Loops

Coding Snippet

```
a = 1
b = 2

while a > b :
        # < repeat block of
        code statements >
```

Text Snippet

```
REPEAT UNTIL a > b
{
< repeat block of code statements
>
}
```

Block Snippet