

# Practice #1

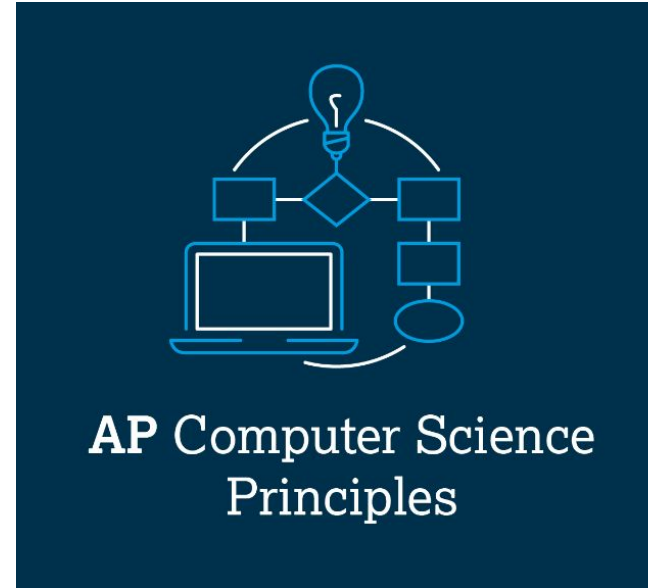
Create Performance Task



# AP CSP Create Performance Task

Part of the AP Exam is to create a program that meets specific requirements:

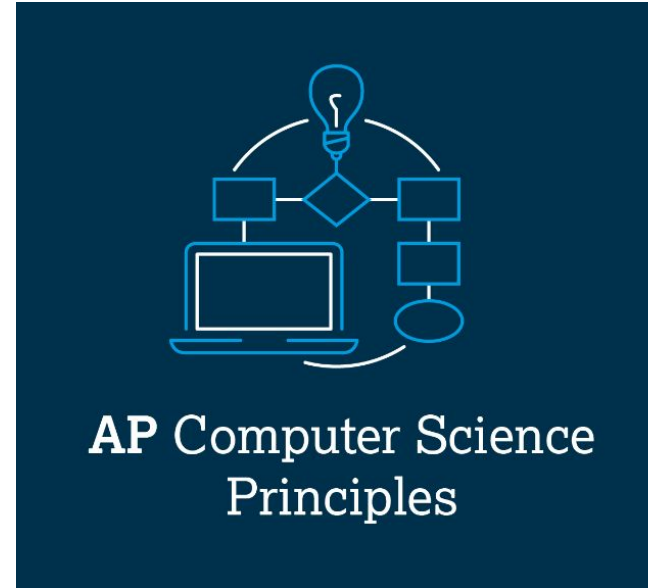
- Creates a list
- Uses a list in a meaningful way
- Has a function with a parameter
- Function has:
  - If statement
  - Loop



# AP CSP Create Performance Task

Things to know about the Create PT:

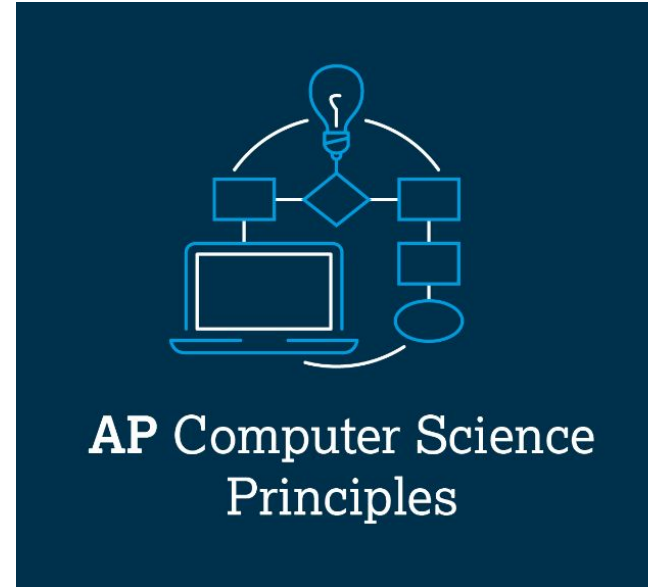
- You can work with a partner
- Your teacher cannot help you
- You cannot use an assigned program for your project, but it can be similar
- Your classmates can help you
- You will be given 9 hours of class time to work on the project



# AP CSP Create Performance Task

Today's program will walk through a practice project that will meet most of the requirements.

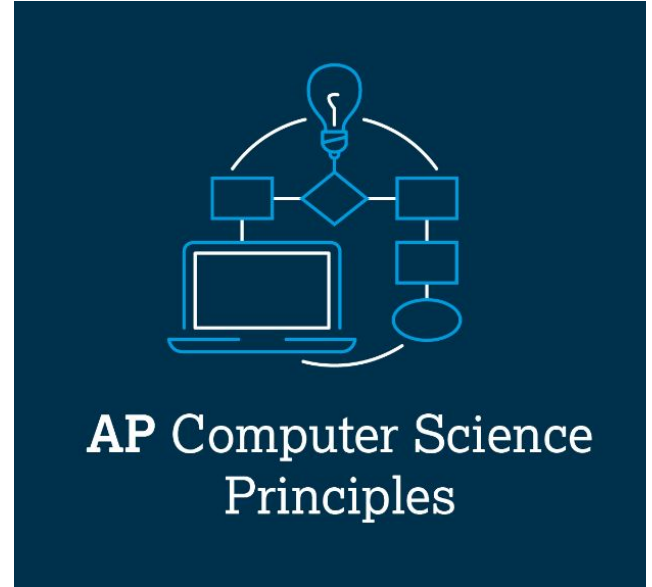
- Start small and add components
- Start with what you know – don't try to do something you don't know how to program for the project
- This isn't a time to learn – it is a time to show what you know how to do



# AP CSP Create Performance Task

For this project, you will focus on:

- Creating and using multiple lists
- Creating a function with a parameter and if statement



# Step #1

## Start a new project

- In CodeSpace, go to Mission 8, objective 9
  - The project will be similar to Mission 7 and Mission 8
- Give your project a descriptive name
  - For today's project, you can call it Practice\_PT\_1
- Use a comment block at the top to include your name, date, partner, description of project, etc.
- Think about any modules you will need to import
  - Codex
  - Random



# Step #2

**Step #2** – Decide on a topic for your first two lists. Since the Diamondbacks are in the World Series, I selected the team as a topic. Then I created a list for the position and another list for the players. You will do something similar, but you can pick your own topic. Make sure you match the information in the two lists. Suggestions: Player/position, musician/song, artist/painting, author/book, family member/name, class/period or teacher. The list should have at least 6 items, but you can have more.

Example of two lists on one topic:

```
dbacks_pos = ["pitcher", "catcher", "1st Base",  
             "2nd Base", "3rd Base", "shortstop"]  
dbacks_player = ["Merrill Kelley", "Gabriel Moreno", "Christian Walker",  
                "Ketel Marte", "Evan Longoria", "Geraldo Perdomo"]
```

Name of list	Data in the list

## Decide on a topic for your project

This project will be about displaying information about a topic.

- Use your assignment document to see an example of two matching lists.
- Fill out the information on your document.
- Then add the two lists to your project.



# Step #3

## Create an intro function

It is always nice to include instructions with your code, and a message when the program ends.

- Create an intro() function that gives general instructions about the program
- Use display.print statements
- Instructions to give:
  - Right button to scroll forward
  - Left button to scroll backward
  - Up button to get a random item
  - Down button to quit





# Step #3

## Create an ending function

It is also nice to include a message when the program ends, so the user knows it has ended.

- Create an ending() function that gives lets the user know the program has ended
- Use display.print statements



# Step #3

## Create an intro and ending function

Your code could look similar to this:

```
def intro():
    display.print("Welcome to the ")
    display.print("D-Backs roster")
    display.print("R = Scroll forward")
    display.print("L = Scroll backward")
    display.print("U = Random player")
    display.print("D = Quit")

def ending():
    display.clear()
    display.print("Thank you!")
    display.print("Have a good day!")
```



# Step #4

## Start your main program

- Initialize an index variable
  - Will be used to scroll left and right
- Call the intro() function
- Start the while True loop

```
# Main Program
intro()
index = 0

while True:
    if buttons.was_pressed(BTN_R):
```



# Step #4

Program the  
“if buttons.was\_pressed”  
for each of the buttons

- Refer to your previous programs (Mission 7 in particular) if you don't remember how to scroll.
- When printing the information, you will use the index for BOTH lists.

```
display.clear()
display.print(dbacks_pos[index])
display.print(dbacks_player[index])
```



# Step #4

**Hint!** Use a different index variable for the random selection:

```
if buttons.was_pressed(BTN_U):
    display.clear()
    rindex = random.randrange(0, len(dbacks_pos))
    display.print(dbacks_pos[rindex])
    display.print(dbacks_player[rindex])
```

## Stop and test frequently!

- Don't try to write all the code at one time
- Mistakes happen, so find them early
- Type just a few lines of code and then run the program
- Test each button press as you code it
- If there is an error, fix it before continuing



# Step #4

## End the program

- Outside the while True loop, call the ending function
- Be careful of your indenting! It needs to be AFTER the loop, not inside

```
if buttons.was_pressed(BTN_D):  
    break  
  
ending()
```



# Step #5

## Add another topic

At this point you should have a working program. However, it only meets half the requirements. You don't have a function with a parameter or if statement yet.

- Think of another topic you can add to the program
- It can be:
  - Another sports team/player
  - A different topic than the first one
    - For example, if my first topic was musicians, my second topic could be theme parks



# Step #5

## Decide on a second topic for your project

- Use your assignment document to see an example of two more matching lists.
- Fill out the information on your document.
- Then add the two additional lists to your project.

```
rangers_pos = ["catcher", "3rd base", "1st base",  
              "shortstop", "2nd base", "outfielder"]  
rangers_player = ["Mitch Garver", "Josh Jung", "Nathaniel Lowe",  
                 "Corey Seager", "Marcus Semian", "Adolis Garcia"]
```





# Step #6

## Creating a function with a parameter

Now the coding for the second topic will be exactly the same as the first topic. The only difference is the names of the lists.

So instead of doing a lot of complicated if statements in the main program, we will create a function with a parameter!



# Step #6

```
def ending():  
    display.clear()  
    display.print("Thank you!")  
    display.print("Have a good day!")  
  
def display_info():  
    |
```

## Make a new function

The first three if statements for button presses use the lists. This is the code that we need to be able to use for either list.

- Near the top of your code, under the intro() and ending() functions, create a new function
- Call it display\_info()



# Step #6

## Make a new function

From your main program, cut the code for the left, right, and up if statements and paste it into your new function



```
def display_info():
    if buttons.was_pressed(BTN_R):
        display.clear()
        display.print(dbacks_pos[index])
        display.print(dbacks_player[index])
        index = index + 1
        if index >= len(dbacks_pos):
            index = 0

    if buttons.was_pressed(BTN_L):
        display.clear()
        display.print(dbacks_pos[index])
        display.print(dbacks_player[index])
        index = index - 1
        if index < 0:
            index = len(dbacks_pos) - 1

    if buttons.was_pressed(BTN_U):
        display.clear()
        rindex = random.randrange(0, len(dbacks_pos))
        display.print(dbacks_pos[rindex])
        display.print(dbacks_player[rindex])
```

# Step #6

## Make a new function

This function uses and changes the variable “index”. Therefore, it needs to be a global. You will learn more about “global” in a future lesson.



```
def display_info():
    global index
    if buttons.was_pressed(BTN_R):
        display.clear()
        display.print(dbacks_pos[index])
        display.print(dbacks_player[index])
        index = index + 1
        if index >= len(dbacks_pos):
            index = 0

    if buttons.was_pressed(BTN_L):
        display.clear()
        display.print(dbacks_pos[index])
        display.print(dbacks_player[index])
        index = index - 1
        if index < 0:
            index = len(dbacks_pos) - 1

    if buttons.was_pressed(BTN_U):
        display.clear()
        rindex = random.randrange(0, len(dbacks_pos))
        display.print(dbacks_pos[rindex])
        display.print(dbacks_player[rindex])
```

# Step #6

```
# Main Program
intro()
index = 0

while True:
    display_info()

    if buttons.was_pressed(BTN_D):
        break

ending()
```

## Make a new function

You don't need to include the "kill switch" because it is not dependent on which lists to print.

- In the while loop, call the `display_info()` function.
- Test your program. It should run exactly the same as it did before.
- If you have any errors, fix them before you continue.



# Step #7

Example:

```
def intro():  
    display.print("Welcome to the ")  
    display.print("D-Backs roster")  
    display.print("A = Diamondbacks")  
    display.print("B = Rangers")  
    display.print("")  
    display.print("R = Scroll forward")  
    display.print("L = Scroll backward")  
    display.print("U = Random player")  
    display.print("D = Quit")
```

## Let the user choose the topic

Now we need to let the user choose which topic to display information about.

- Use the A and B buttons for user choice
- Add print statements to your intro letting the user know to select a topic by choosing either A or B



# Step #7

## Let the user choose the topic

- Initialize “state” to 1
- Add if statements to the infinite loop that will set a “state” variable to either 1 or 2
- Include the “state” variable as an argument in the call statement

```
# Main Program
intro()
index = 0
state = 1

while True:
    if buttons.was_pressed(BTN_A):
        state = 1

    if buttons.was_pressed(BTN_B):
        state = 2

    display_info(state)

    if buttons.was_pressed(BTN_D):
        break

ending()
```





# Step #7

## Use the “state” variable in the function

- Add “state” as a parameter
- Add an if statement that will set a new local list variables to the chosen topic

```
def display_info(state):  
    global index  
    if state == 1:  
        the_list1 = dbacks_pos  
        the_list2 = dbacks_players  
    else:  
        the_list1 = rangers_pos  
        the_list2 = rangers_players  
  
    if buttons.was_pressed(BTN_R):  
        display.clear()  
        display.print(the_list1[index])  
        display.print(the_list2[index])  
        index = index + 1  
        if index >= len(the_list1):  
            index = 0
```





# Step #7

## Use the new local lists in the function

- Go through the code in the function and replace your original lists with the new local lists
- Make sure you go through all the if statements

```
if buttons.was_pressed(BTN_R):
    display.clear()
    display.print(the_list1[index])
    display.print(the_list2[index])
    index = index + 1
    if index >= len(the_list1):
        index = 0

if buttons.was_pressed(BTN_L):
    display.clear()
    display.print(the_list1[index])
    display.print(the_list2[index])
    index = index - 1
    if index < 0:
        index = len(the_list1) - 1

if buttons.was_pressed(BTN_U):
    display.clear()
    rindex = random.randrange(0, len(the_list1))
    display.print(the_list1[rindex])
    display.print(the_list2[rindex])
```



# Step #8

## Test and debug

- Run your code for accuracy and bugs. Try both A and B buttons while the code is running.
- Does the information displayed match the button selected?
- Make sure you test all buttons thoroughly.



# Challenge!

## Add the topic

- The information is printed, but does the user remember what topic is displayed?
- Add a variable for the topic, and assign it a value when assigning the local list variables
- Print the topic before each piece of information



# And now you have your own create PT practice

## Congratulations!

By completing this practice project you have prepared for the PT by:

- Creating a list
- Using the list in a meaningful way
- Creating a function with a parameter
- Calling the function
- Using sequence and selection in the function
- Using the parameter in an if statement



# And now you have your own create PT practice

## Moving forward

You will continue to prepare for the Create PT by:

- Using iteration in the function
- Learning about global and local variables
- Learning when a parameter is needed

