AP CSP Python with CodeX **PPR Code Segment Activity Guide**	**Name:**

Introduction

For this assignment, you will practice identifying parts of your code that are required for the Create PT and creating images of the code that you can upload into the digital portfolio. For practice, you will insert them in this assignment document instead of the digital portfolio.

Warm-Up

What are the requirements for the Create PT?	● Create a list and use it in a meaningful way ● Has a function with a parameter that affects function execution ● Function has a loop and if statement
What are some things you can and cannot do during the Create PT?	Answers will vary. Possible answers: Can ● Work with a partner ● Get help with code from classmates ● Remix a project Cannot ● Use an assignment as the project – needs to be original ● Be assigned a project by the teacher ● Ask the teacher for help ● Get help during the PDF, video and PPR

PPRs for Create PT practice programs

The forms below represent information needed to upload in the AP Digital Portfolio. The instructions are given, with a box for uploading images.
- On a Windows PC or laptop, use the snipping tool to capture parts of your code.
- Save the code segment as a PNG file. Give it a descriptive name
- Save it where you can find it later for uploading.
- Follow the instructions on the slides to save the code segments for PT_Practice1.
- Then go to the first form on the next page and insert the images by uploading them from your computer.
- You will see the exact words from the digital portfolio.
- Continue the Activity Guide by creating a PPR for other practice Create PT programs.

Things to check for:
- Did the student change the background to light?
- Did the student remove ALL comments?
- Did the student create a snippet of the correct part of the project?
- Are the snippets in the correct boxes?

The solutions shown aren't the only solutions. For example, if there is more than one list, students can choose either list.

Now is the time to give feedback, before they start the Create PT project.

Personalized Project Reference – PT_Practice1

Procedure: Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

i. The first program code segment must be a student-developed procedure that:

  □Defines the procedure's name and return type (if necessary)

  □Contains and uses one or more parameters that have an effect on the functionality of the procedure

  □Implements an algorithm that includes sequencing, selection, and iteration

```python
def turn_pixels(choice):
    if choice == 1:
        the_list = pix_list1
    else:
        the_list = pix_list2
    for index in range(len(the_list)):
        for pix in range(4):
            pixels.set(pix, the_list[index][0], the_list[index][1])
        sleep(delay)
```

ii. The second program code segment must show where your student-developed procedure is being called in your program.

```python
while True:
    if buttons.was_pressed(BTN_A):
        turn_pixels(1)

    if buttons.was_pressed(BTN_B):
        turn_pixels(2)
```

List: Capture and paste two program code segments you developed during the administration of this task that contain a list (or other collection type) being used to manage complexity in your program.

i. The first program code segment must show how data has been stored in the list.

```python
pix_list2 = [ [RED, 50], [GREEN, 50], [BLUE, 75],
              [YELLOW, 100], [PINK, 40], [ORANGE, 80],
              [CYAN, 60], [PURPLE, 50], [WHITE, 25] ]
```

ii. The second program code segment must show the data in the same list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

Note: You can use the same snippet for the function and the list, if the function uses the list.

```python
def turn_pixels(choice):
    if choice == 1:
        the_list = pix_list1
    else:
        the_list = pix_list2
    for index in range(len(the_list)):
        for pix in range(4):
            pixels.set(pix, the_list[index][0], the_list[index][1])
        sleep(delay)
```

Personalized Project Reference – PT_Practice2

Procedure: Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

i. The first program code segment must be a student-developed procedure that:

       □Defines the procedure's name and return type (if necessary)

       □Contains and uses one or more parameters that have an effect on the functionality of the procedure

       □Implements an algorithm that includes sequencing, selection, and iteration

```python
def play_game(level):
    if level == 1:
        delay = 1.5
        tries = 5
    elif level == 2:
        delay = 1.0
        tries = 10
    else:
        delay = 0.5
        tries = 15
    for count in range(tries):
        item = random.choice(button_list)
        message = item[0]
        button = item[1]
        display.show(message)
        sleep(delay)
        pressed = buttons.is_pressed(button)
        if pressed:
            pixels.fill(GREEN)
        else:
            pixels.fill(RED)
        # clear button presses
        buttons.was_pressed()
        time.sleep(1)
```

ii. The second program code segment must show where your student-developed procedure is being called in your program.

```python
while True:
    if buttons.was_pressed(BTN_L):
        play_game(1)
    if buttons.was_pressed(BTN_U):
        play_game(2)
    if buttons.was_pressed(BTN_R):
        play_game(3)
```

List: Capture and paste two program code segments you developed during the administration of this task that contain a list (or other collection type) being used to manage complexity in your program.

i. The first program code segment must show how data has been stored in the list.

```
button_list = [ ["Hold BTN-A", BTN_A], ["Hold BTN-B", BTN_B],
                ["Hold UP", BTN_U], ["Hold DOWN", BTN_D],
                ["Hold RIGHT", BTN_R], ["Hold LEFT", BTN_L] ]
```

ii. The second program code segment must show the data in the same list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

Note: The entire function can be used, or the part of the function that uses the list:

```
for count in range(tries):
    item = random.choice(button_list)
    message = item[0]
    button = item[1]
    display.show(message)
    sleep(delay)
    pressed = buttons.is_pressed(button)
    if pressed:
        pixels.fill(GREEN)
    else:
        pixels.fill(RED)
```

Personalized Project Reference – PT_Practice3

Procedure: Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

i. The first program code segment must be a student-developed procedure that:
  □Defines the procedure's name and return type (if necessary)
  □Contains and uses one or more parameters that have an effect on the functionality of the procedure
  □Implements an algorithm that includes sequencing, selection, and iteration

```python
def pick_student(period, count):
    delay = 0.0
    if period == 1:
        the_list = peanuts
        class_name = "Peanuts"
    else:
        the_list = simpsons
        class_name = "Simpsons"

    for number in range(count):
        student = random.choice(the_list)
        display.clear()
        display.print(class_name, scale=4)
        display.print(student, color=YELLOW, scale=4)
        delay = delay + 0.01
        time.sleep(delay)
```

ii. The second program code segment must show where your student-developed procedure is being called in your program.

```python
while True:
    if buttons.was_pressed(BTN_A):
        count = random.randrange(10, 20)
        pick_student(1, count)

    if buttons.was_pressed(BTN_B):
        count = random.randrange(10, 20)
        pick_student(2, count)
```

List: Capture and paste two program code segments you developed during the administration of this task that contain a list (or other collection type) being used to manage complexity in your program.
i. The first program code segment must show how data has been stored in the list.

```python
peanuts = ["Snoopy", "Woodstock", "Charlie", "Linus", "Sally",
           "Lucy", "Franklin", "Pigpen", "Violet"]
```

ii. The second program code segment must show the data in the same list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

```python
def pick_student(period, count):
    delay = 0.0
    if period == 1:
        the_list = peanuts
        class_name = "Peanuts"
    else:
        the_list = simpsons
        class_name = "Simpsons"

    for number in range(count):
        student = random.choice(the_list)
        display.clear()
        display.print(class_name, scale=4)
        display.print(student, color=YELLOW, scale=4)
        delay = delay + 0.01
        time.sleep(delay)
```

Personalized Project Reference – PT_Practice4

Procedure: Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

i. The first program code segment must be a student-developed procedure that:

       □Defines the procedure's name and return type (if necessary)

       □Contains and uses one or more parameters that have an effect on the functionality of the procedure

       □Implements an algorithm that includes sequencing, selection, and iteration

```python
def play_game(speed):
    if speed == 'fast':
        delay = 0.5
    else:
        delay = 0.75

    letter = random.choice(letter_list)
    start_of_game(letter)

    for count in range(100):
        display.clear()
        item = random.choice(letter_list)
        display.print(item, scale=8)
        time.sleep(delay)
        if buttons.was_pressed(BTN_A):
            break

    end_of_game(item, letter)
```

ii. The second program code segment must show where your student-developed procedure is being called in your program.

```python
while True:
    if buttons.was_pressed(BTN_A):
        play_game('fast')
    if buttons.was_pressed(BTN_B):
        play_game('slow')
```

List: Capture and paste two program code segments you developed during the administration of this task that contain a list (or other collection type) being used to manage complexity in your program.

i. The first program code segment must show how data has been stored in the list.

```python
letter_list = ['b','t','c','w','d','r','f','p','g','m','h']
```

ii. The second program code segment must show the data in the same list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

```python
for count in range(100):
    display.clear()
    item = random.choice(letter_list)
    display.print(item, scale=8)
    time.sleep(delay)
    if buttons.was_pressed(BTN_A):
        break
```