**Name:** Aika Manalo

## Instructions

1. Complete the activities during the lecture.
2. Paste your code screenshots and type your answers in the sections below.
3. Save as **Lastname_Firstname_W4_Activity.pdf** and upload.

Activity 1: Scope Analysis

**Which option represents the correct output?**
[ ] A) 5 5
[ ] B) 5 10
[ ] C) 10 100
[ ] D) 5 100

Activity 2: Code Prediction

 **1. What will the code output?**

- Inside: 15

- Outside: 10

**2. Why is the "Outside" value different from the "Inside" value?**
*> C is pass-by-value meaning that the function receives a "copy" of the value and modifies it but, the original variable remains unchanged (untouched).*

**3. Concept Check:** Is there a way to make the change persist outside the function?
*> Yes, by pointers since they are pass-by-reference so we can modify the original address.*

**Activity 3: Modular Construction**

Match the code snippet to the correct file location:

| Code Snippet | Correct File Location |
|---|---|
| **A.** int square(int x); | header |
| **B.** printf(…, square(5)); | main |
| **C.** int square(int x) { return x * x; } | implementation |

**Activity 4: Address Logic**

Based on the memory map provided in the slide:

1. **Value of p:** 0x100
2. **Value of &a:** 0x100
3. **Value of &p:** 0x108

**Activity 5: Mini-Coding**

**Paste your screenshot here:**

```
6      void reset(int *a, int *b)
7      {
8          *a = 0;
9          *b = 0;
```

**Activity 6: Spot the Bug**

Which line of code safely initializes the pointer to fix the crash?

[ ] A) ptr = 0; - set's it to NULL which causes a crash
[ ] B) ptr = &val; - assigns address of val to ptr
[ ] C) *ptr = val; - dereferences pointer and tries to PBV val into mem location ptr is pointing to
[ ] D) &ptr = val; - tries to assign value of val to the address of ptr itself