CS Elective 2 Prelim Lab Exam

PHP Basics (Game Items) | BSCS 2A

September 08, 2025| Lab 06 4PM - 730PM

Instructions:

- Use PHP 7.4+ syntax (short <?php tags).
- Create each file in the same folder and test each by opening it in a PHP-enabled server (XAMPP, MAMP, built-in php -S localhost:8000, etc.).
- **Do not change the variable names or function names** asked for in each task — grading expects those exact names.
- Put comments (//) where asked. Keep code tidy.
- When done, ask for your instructor's attention for checking.

Question 1: Player Profile (10 pts) — player_profile.php

1. Declare a constant named "***GAME_TITLE***" with value ***Dungeon Quest***
2. Declare variables:
   - *playerName* set 'Aria';
   - *playerLevel* set 3
   - *playerHP* set 45
3. Below the PHP block, write simple HTML and print the values. Use either echo with concatenation (.) or shorthand <?= ... ?>. Example output must show: ***Game title, Player name, Level, and HP (e.g. HP: 45 / 100).***
4. Do not add CSS or external files.

Question 2: Enemy Wave (20 pts) — enemy_wave.php

1. Declare variables:
   a. enemyBaseName set Goblin
   b. enemyCount set 10
2. Use a for loop from 1 to enemyCount. For each $i: Print this line

   Goblin: #1
   Goblin #2
   …
   Goblin #10
   Total HP: 100

Note: Total HP is just count * 10

Question 3: Combat Calculator (25 pts)— combat_calculator.php

1. ***Declare the required constant***
   - *On the next line, declare the constant named **MIN_DAMAGE** and set it to **2.** (Use the exact constant name **MIN_DAMAGE**.)*
2. ***Implement the required function (follow these rules exactly)***
   - *Define a function named calculate_damage that accepts two parameters: $attack and $defense.*
   - *Inside the function:*
     - *Compute a local variable (call it $damage) equal to $attack - $defense.*
     - *If $damage is less than MIN_DAMAGE, set $damage to MIN_DAMAGE.*

        - *Return $damage from the function.*
    - ***Important****: use the exact function name calculate_damage and parameter names $attack and $defense.*

3. ***Set the required test variables***
    - *After the function, create two variables with these exact names and values:*
        - *$playerAtk with the value 30*
        - *$enemyDef with the value 18*
4. ***Call the function***
    - *Call calculate_damage passing the variables $playerAtk and $enemyDef, and store the returned value in a variable named $damage.*
5. ***Output the result using concatenation***
    - *Print a single sentence that uses concatenation to show the values and the computed damage. The sentence should look like:*
        - *Player attack: 30 — Enemy defense: 18 — Damage dealt: 12*
    - *Make sure you build that sentence using string concatenation (not plain plain text), and use the variable values so the output will change if the input variables change.*