

**Date: 9/19/2025**

## **Lab Session: Ashesi Premier League – Football Player Management System**

### **Scenario Background**

Ashesi is hosting a **University Football Tournament** called the *Ashesi Premier League*. Each team must register players, track their statistics, and decide who plays based on performance and conditions. You have been asked to build a small Java program that helps the coach make decisions about player eligibility, position, and lineup.

This lab will guide you step by step, starting with simple tasks and ending with a combined program that pulls everything together.

### **Objectives**

- Practice basic Java syntax with **variables**, **data types**, and **initialization**.
- Apply **type casting**, **increment/decrement operators**, and **logical operators**.
- Use selection statements: **if/else**, **nested ifs**, **switch**, and **conditional expressions**.
- Understand and practice **switch fall-through (running cases)**, both as a mistake and as a useful grouping technique.
- Combine all concepts into a single, working player selection simulator.

### **Learning Outcomes**

By the end of this lab, you should be able to:

1. Declare, assign, and initialize variables in Java.
2. Convert between data types using casting.
3. Apply increment (++) and decrement (--) operators.
4. Write boolean expressions and use logical operators (&&, ||, !).
5. Use selection statements (if, else if, else, nested ifs).
6. Implement switch cases with and without break.
7. Use the conditional (ternary) operator.
8. Integrate all concepts into a small application.

### **Concepts to be Practiced**

- Variables and data types
- Type casting
- Increment and decrement operators
- Boolean expressions and logical operators
- If, else-if, and nested if statements
- Switch statements (with break and fall-through)
- Conditional expression (? :)

## Final Results

At the end of the lab, you will have a **Java program** that:

- Accepts player details (name, age, height, weight, jersey number) from the keyboard.
- Converts height to centimeters and rounds weight.
- Categorizes the player as *Rising Star*, *Prime Player*, or *Veteran*.
- Identifies the player's position based on the jersey number.
- Determines whether the jersey number is an *attacker number*.
- Evaluates eligibility using logical conditions.
- Decides whether the player is in the starting lineup (nested if).
- Outputs a structured "Player Report" with a final decision: **Play** or **Rest**.
- Demonstrates how a switch behaves **with missing breaks** (unwanted fall-through) and how to use **grouped cases** correctly.

## Tasks

### Task 1 – Declaring and Using Variables

**Description:** Create variables to represent a player's details: name, age, height, weight, and jersey number.

**Objective:** Practice variable declaration, assignment, and initialization.

**Concepts to Practice:** Variables, data types, initialization.

#### Instructions:

- Create a new Java project called Lab-01-Cohort-(add your cohort).
- Create a new java class named `FootballPlayer`.
- Declare variables for the player details: name, age, height (in meters), weight (in pounds), and jersey number. **Ensure you assign the appropriate data type to each variable declared.**
- Using the `Scanner` class, ask the user to enter values for the player details.
- Print the details on the screen.
- Sample Output:
  - Player Name – Kwesi Kwabena Ekow
  - Age – 25
  - Height – 1.75m
  - Weight – 135lbs
  - Jersey Number - 6

## Task 2 – Constants, Data Types and Type Casting

**Description:** Convert the player's height to centimeters, weight to kilogram, and round weight to an integer.

**Objective:** Practice type casting and conversions.

**Concepts to Practice:** Type casting, primitive data types.

### Instructions:

- Create two constant variables, `pound` and `meter` and set their values to `0.45359237` and `100` respectively.
- Convert the weight to kilograms and the height to centimeters.
- Round the converted weight in kilograms to integer by casting. Don't use any rounding method. Use casting for this task.
- Display details of the player.
- Sample output:
  - Player Name – Kwesi Kwabena Ekow
  - Age – 25
  - Height in centimeters – 175cm
  - Weight in kilograms– 61kg
  - Jersey Number - 6

## Task 3 – Increment and Decrement Operators

**Description:** Simulate changes in the player's statistics over time. A player's age increases by 1 after each season.

**Objective:** Apply increment and decrement operators.

**Concepts to Practice:** `++`, `--`.

### Instructions:

- Print the player's age now.
- Use `++` to increase age by 1.
- Simulate a penalty where the jersey number decreases by 1.
- Print the updated values – the new age jersey number.

## Task 4 – Boolean Expressions and Logical Operators

**Description:** The coach wants to check if a player is eligible to play. The rules are:

- The player must be at least 18 years old.
- The player must not be older than 35.
- The player's weight must be less than 90 kg.

**Objective:** Write and evaluate boolean expressions.

### Concepts to Practice:

- Relational operators ( $\geq$ ,  $\leq$ ,  $<$ )
- Logical AND ( $\&\&$ ), OR ( $\|\|$ ), NOT ( $!$ )

### Instructions:

1. **Check normal eligibility with  $\&\&$ :**
    - a. Write boolean expression for eligibility.
    - b. If this is true, print "Eligible". Otherwise, print "Not Eligible".
  2. **Experiment with OR ( $\|\|$ ):**
    - a. Write a condition to check if a player is **either underage or overweight**:  
Rule: player is underage when they are less than 18 years. Overweight when weight is greater or equal to 90.
    - b. If this condition is true, print "Player has a problem (either too young or too heavy)".
  3. **Experiment with NOT ( $!$ ):**
    - a. Take the original eligibility condition in 1a.
    - b. Apply NOT to it
    - c. Print "Not Eligible" if the NOT condition is true.
    - d. Test with values that make the player eligible (e.g., age = 25, weight = 75) and observe how the NOT flips the result.
- 

### Task 5 – Selection Statements (if-else)

**Description:** Categorize players by age:

- Under 20  $\rightarrow$  *Rising Star*
- 20–30  $\rightarrow$  *Prime Player*
- Over 30  $\rightarrow$  *Veteran*

**Objective:** Practice if-else statements.

**Concepts to Practice:** If, else if, else.

### Instructions:

- Write conditions for each category.
- Print the player's category.

## Task 6 – Switch Cases (Basic)

**Description:** Assign position based on jersey number.

**Objective:** Use a switch statement with proper breaks.

**Concepts to Practice:** switch with case, default, break.

### Instructions:

- Ask the coach to enter the jersey number of the player.
- Write a switch using the jersey number entered (e.g., 1 = Goalkeeper, 2= defender, 6 = midfielder, 7 = Winger, 9 = Striker, 10 = Playmaker, 11 = Winger, 5- defender, 8 = midfielder). Each case should print the player position.
- When the jersey number entered does not match any case, print “Player position not known”

## Task 7 – Switch Cases (Running Case / Fall-Through)

**Description:** Explore how cases behave with and without break.

**Objective:** Understand fall-through in switch.

**Concepts to Practice:** Switch without break, grouped cases.

### Instructions:

- **Part A (Unwanted Fall-Through):** Write a switch on jersey number. Remove breaks for cases 2,6, and 7 and observe multiple lines printing. Don’t forget to include a default case.
- **Part B (Grouped Case – Useful Fall-Through):** When we observe the switch cases in Task 6, we see some of the cases can be grouped because they print the same thing. Write a new switch case, grouping the cases that are the same and printing the player position. Don’t forget to include a default case.

## Task 8 – Nested Ifs

**Description:** Select player for starting lineup. Rule: if category is *Prime Player* and weight < 80, then starting lineup; else bench.

**Objective:** Practice nested decision making.

**Concepts to Practice:** Nested if statements.

### Instructions:

- Write a selection to first check category first.
- Inside the first check, add a selection to check weight.
- Print lineup decision.

## Task 9 – Conditional Expression (Ternary Operator)

**Description:** Using the eligibility check in task 1, quickly decide final status: if eligible, **Play**, otherwise **Rest**.

**Objective:** Use ternary operator.

**Concepts to Practice:** Conditional operator ? :.

### Instructions:

- Write a ternary expression for final decision using the eligibility condition in Task 4.
  - Print result.
- 

## Task 10 – Combine Everything: Team Selection Simulator

**Description:** Build a complete program that integrates all tasks into one flow.

**Objective:** Apply all concepts in a single Java program.

**Concepts to Practice:** All concepts (variables, casting, increment/decrement, boolean logic, if/else, nested ifs, switch, fall-through, ternary).

### Instructions:

- Create a new project named **Ashesi Premier League**
- Create a java class called **PlayerSelection**.
- Use Scanner to read player details from the keyboard.
- Convert data where needed.
- Determine category, position, attacker status, eligibility.
- Make lineup and final decision.
- Print a **Player Report** showing:
  - Player: [Name]
  - Age: [Age] ([Category])
  - Height: [cm]
  - Weight: [kg]
  - Jersey: [number]
  - Position: [Position]
  - Attacker jersey: [Yes/No]
  - Eligibility: [Eligible/Not Eligible]
  - Lineup Decision: [Decision]
  - Final Decision: [Play/Rest]
- Run the unwanted fall-through demo to observe the difference.