Name: _____ Session: AM / PM

Circle one

Chapter 3 - Day 1

Mission Readiness

DAY 1 FOUNDATIONS

How decisions work: A condition is an expression that is either true or false. When a condition is true, the code inside the braces runs; when it is false, that code is _____. An _____ selects exactly one of two alternatives.

Term	Definition / memory cue
condition	
boolean	
relational operator	
flag	
if	
if-else	

Term	Definition / memory cue
nested if	
if-else-if chain	
trailing else	
format specifier	

Relational Operators

Operator	Mission example	Meaning
`<`	`fuel < 25`	
`<=`	`fuel <= 25`	
`>`	`battery > 60`	
`>=`	`battery >= 60`	
`==`	`crew == 4`	
`!=`	`crew != 0`	

= versus ==: A single = _____ a value to a variable. A double == _____ two values and produces true or false.

DEMONSTRATION: MISSION READINESS

Mission rules: These are classroom rules, not real spacecraft rules. Fuel below 25 gives a low-fuel warning. A battery of at least 60 meets the battery target. The mission is ready only when the battery target is met and fuel is at least 25. Fuel below 25 is Low, 25 up to (but not including) 75 is Reserve, and 75 or above is Ample.

E1-A Input before logic

`MissionReadinessStep1.java`

What changed: Reads fuel and battery percentages and echoes them back, so we can confirm the input is stored correctly before adding any decisions.

```java
import java.util.Scanner;

public class MissionReadinessStep1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter fuel percentage: ");
        double fuel = sc.nextDouble();
        System.out.print("Enter battery percentage: ");
        double battery = sc.nextDouble();
        System.out.println("Fuel: " + fuel + ", Battery: " + battery);
    }
}
```

1. Variable that stores the fuel value:

2. Data type of battery:

3. Last line of input code (decisions will be added after it): line

E1-B Plain if, a flag, if-else, and a nested check

`MissionReadinessStep2.java`

What changed: Replaces the echo with a low-fuel warning (plain if), a battery flag with an if-else, and a readiness check nested inside the battery-ready path.

```
11          // Plain if: prints only when the condition is true
12          if (fuel < 25) {
13              System.out.println("WARNING: Low fuel!");
14          }
15
16          // Flag + if-else, with a nested if on the true path
17          boolean batteryReady = (battery >= 60);
18          if (batteryReady) {
19              if (fuel >= 25) {
20                  System.out.println("Mission ready.");
21              }
22          } else {
23              System.out.println("Battery not ready.");
24          }
```

1. If fuel is 20, does line 13 run?

2. Value stored in batteryReady when battery is 60:

3. The inner if on line 19 is checked only when

4. Why can 'Mission ready.' and 'Battery not ready.' never both print?

Predict the full output before the program is run. Write every line that prints; write (nothing) if no line prints.

Fuel	Battery	Output	Ran it?
24.9	70		☐
25	60		☐
20	50		☐
74.9	59.9		☐

E1-C Fuel category chain

MissionReadinessStep3.java

What changed: Adds an if-else-if chain that places fuel into exactly one category: Low, Reserve, or Ample. Lines 1-25 are unchanged from Step 2.

```
26          // NEW in Step 3: if-else-if chain; first true branch wins
27          if (fuel < 25) {
28              System.out.println("Category: Low");
29          } else if (fuel < 75) {
30              System.out.println("Category: Reserve");
31          } else {
32              System.out.println("Category: Ample");
33          }
```

1. If fuel is 75, the category printed is

2. When line 29 is reached, fuel must be

3. If `fuel < 75` were checked first, fuel = 10 would print

4. The trailing else on line 31 runs when

Predict the category for each fuel value. Watch the boundaries.

Fuel	Category	Ran it?
`24.9`		☐
`25`		☐
`74.9`		☐
`75`		☐

E1-D Readable decimals

`MissionReadinessStep4.java`

What changed: Adds a two-decimal fuel display after the category. The stored value and every comparison stay exactly the same.

```
26          // if-else-if chain; first true branch wins
27          if (fuel < 25) {
28              System.out.println("Category: Low");
29          } else if (fuel < 75) {
30              System.out.println("Category: Reserve");
31          } else {
32              System.out.println("Category: Ample");
33          }
34
35          // NEW in Step 4: display only; fuel itself is unchanged
36          System.out.printf("Fuel: %.2f%%%n", fuel);
```

1. `%.2f` displays a double with
2. `%%` prints
3. `%n` prints
4. Why does 74.999 display as 75.00 but still print Reserve?

Battery is 80 for every row. Predict the category line and the display line.

Fuel	Category / Display	Ran it?
`74.999`		☐
`75`		☐
`24.999`		☐