Name: _____ Session: AM / PM
Circle one

Chapter 3 - Day 3

Exact Choices and Formatted Output

DAY 3 FOUNDATIONS

How a switch works: A switch compares one value, the selector, against exact case labels. Execution starts at the matching label and keeps going until it reaches a _____. The default section runs when _____.

Term	Definition / memory cue
selector	
case label	
break	
default	
fall-through	
stacked labels	

switch or if-else-if?: Use a switch for _____ of one selector, such as menu choices 1 to 5. Use an if-else-if chain for _____, such as fuel below 25. Case labels must be compile-time constants, and the selector can be an int, a char, or a String, but not a double.

Term	Definition / memory cue
format specifier	
flag	
field width	
precision	
String.format	

Format Specifier Reference

Specifier	Example value	What it does
`%d`	`42`	
`%05d`	`42`	
`%.2f`	`3480.5`	
`%,.2f`	`3480.5`	
`%-10s`	`"Food"`	
`%n`		

Specifier order: The parts always go in this order: % then _____, then width, then precision, then the conversion character. `%,12.2f` is correct; `%.2,12f` is not.

DEMONSTRATION: MISSION CARGO CONSOLE

Mission rules: Cargo choices are 1 Water ($1,250.00), 2 Oxygen ($3,480.50), 3 Food ($2,215.75), and 4 Tools ($860.00). Any other number is Unknown with a cost of $0.00. These are classroom values.

E3-A The cargo menu

`MissionCargoStep1.java`

What changed: Shows the menu, reads the choice, and echoes it, so we know the selector holds the right value before we switch on it.

```
5         Scanner keyboard = new Scanner(System.in);
6         System.out.println("1 Water  2 Oxygen  3 Food  4 Tools");
7         System.out.print("Enter a cargo choice: ");
8         int choice = keyboard.nextInt();
9         System.out.println("You chose " + choice);
```

1. The selector for our switch will be

2. Data type of choice:

E3-B Exact choices with switch

MissionCargoStep2.java

What changed: Replaces the echo with a switch that sets the cargo name and cost for each exact choice.

```
10          // Exact choices: one selector, one case per value
11          String cargo;
12          double cost;
13          switch (choice) {
14              case 1:
15                  cargo = "Water";
16                  cost = 1250.00;
17                  break;
18              case 2:
19                  cargo = "Oxygen";
20                  cost = 3480.50;
21                  break;
22              case 3:
23                  cargo = "Food";
24                  cost = 2215.75;
25                  break;
26              case 4:
27                  cargo = "Tools";
28                  cost = 860.00;
29                  break;
30              default:
31                  cargo = "Unknown";
32                  cost = 0.0;
33          }
34          System.out.println("Cargo: " + cargo);
```

1. Choice 2 starts running at line

2. Line 21 stops the switch with

3. Choice 9 runs the _____ section

4. Why does line 34 compile?

Predict the Cargo line.

Choice	Output	Ran it?
2		☐
4		☐
9		☐

E3-C Stacked labels and fall-through

MissionCargoStep3.java

What changed: Reads a destination and switches on a String. Luna and Moon share one route.

```
36          // Stacked labels: Luna and Moon share one instruction
37          keyboard.nextLine(); // throw away the leftover newline
38          System.out.print("Enter the destination: ");
39          String destination = keyboard.nextLine();
40          switch (destination) {
41              case "Luna":
42              case "Moon":
43                  System.out.println("Route: lunar transfer");
44                  break;
45              case "Mars":
46                  System.out.println("Route: Mars transfer");
47                  break;
48              default:
49                  System.out.println("Route: check with Mission Control");
50          }
```

1. Why is line 37 needed?

2. Case "Luna" has no statements, so it _____ to the Moon instructions

3. Typing moon (lowercase) runs the _____ section

4. Demo: with the break on line 17 removed, choice 1 prints

Predict the Route line.

Destination	Output	Ran it?
`Moon`		☐
`Luna`		☐
`moon`		☐
`Mars`		☐

E3-D The formatted cargo report

MissionCargoStep4.java

What changed: Prints a zero-padded mission ID and an aligned cargo line, then builds the same information with String.format and prints it later.

```
52          // Formatted report: printf prints now, String.format stores
53          int missionId = 42;
54          System.out.printf("Mission %05d%n", missionId);
55          System.out.printf("|%-10s|%,12.2f|%n", cargo, cost);
56          String report = String.format("%s cargo: $%,.2f", cargo, cost);
57          System.out.println("Report saved.");
58          System.out.println(report);
```

1. Line 54 displays

2. In `%,12.2f`, the 12 is the

3. Line 56 does not print anything because String.format

Choice 2, destination Mars. Predict the last four lines. Write spaces as they appear.

Line	Output	Ran it?
54		☐
55		☐
57		☐
58		☐

ERROR CLINIC

Three kinds of failure: A compile-time error stops the program from building. A run-time error crashes it while it runs. A _____ runs normally but does the wrong thing.

Fix it: A semicolon after the condition

Broken on purpose

```
1 if (cost > 1000);
2     System.out.println("Heavy cargo fee applies.");
```

Cause:

Repair:

When does it fail?

Fix it: Comparing Strings with ==

Broken on purpose

```
1 if (destination == "Mars") {
2     System.out.println("Route: Mars transfer");
3 }
```

Cause:

Repair:

When does it fail?

Fix it: Precision on a whole number

Broken on purpose

```
1 System.out.printf("Mission %.2d%n", missionId);
```

Cause:

Repair:

When does it fail?

Fix it: An expression as a case label

Broken on purpose

```
switch (choice) {
    case choice > 4:
        System.out.println("Invalid");
        break;
}
```

Cause:

Repair:

When does it fail?