Tutorial Sheet - Topic 09

Programming Fundamentals 1

Product Validation, Loops, Switch Statements, and Menus

1. Product Class & Validation (Shop 2.2)

Q1 -- Getter Methods

Given the following field:

```
private double unitCost;
```

Write the getter method for `unitCost`.

Q2 -- Setter Methods

Write the setter method for `unitCost`. You do **not** need to include validation yet.

Q3 -- toString()

Explain what `toString()` should return in an object-oriented program. Why is `toString()` especially useful when debugging?

Q4 -- Apply Validation

Write a **setter method** for `unitCost` with the following rule:

- Valid unit cost: **1.00 to 5000.00**

Q5 -- String Validation

The product name must be **max 20 characters**.

Given the input:

```
"SuperUltraWideMonitorDisplay"
```

State what the stored value should be after the constructor validation.

---

2. Loops (Counter-Controlled, Sentinel, Flag-Based)

Q6 -- Sentinel-Based Loop (Simple)

Write a loop that repeatedly reads integers until the user enters `-1`.
Print the total sum at the end.

(No array needed.)

Q7 -- Counting Inputs

Extend your previous solution so it also prints **how many** numbers were entered before `-1`.

Q8 -- Flag Example

Given an array of integers, write pseudocode that checks if **any** number is odd.
If at least one is odd → print a message.
If none are odd → print a different message.

Q9 -- Boolean Method Version

Rewrite Q8 as a **method**:

```
public boolean containsOdd(int[] nums)
```

Method returns `true` if an odd number exists. (Assume all elements of array are populated.)

Q10 -- Loop Control Variable (LCV)

Briefly explain the term **Loop Control Variable** using your own words.

Q11 -- Error Spotting

Explain what is wrong with this sentinel loop:

```
int num = input.nextInt();
int sum = 0;

while(num != -1){
    sum += num;
}
```

Provide a corrected version.

3. Switch Statement

Q12 -- Switch vs If-Else

Describe one advantage of using a `switch` statement instead of a long chain of `if/else if`.

Q13 -- Basic Switch

Write a switch that prints:

- "Hello" if `choice == 1`
- "Goodbye" if `choice == 2`
- "Invalid option" otherwise

---

4. Menu-Driven Applications

Q14 -- Menu Concept

What is a **menu-driven program**?
Give an example of when you would use one.

Q15 -- Simple Menu Output

Write the code to print the following menu using a raw string (triple quotes):

```
1. Add Product
2. View Product
3. Exit
```

Q16 -- Switch + Menu

Write a switch on `option` that:

- Calls `addProduct()` for case 1\
- Calls `viewProduct()` for case 2\
- Prints "Exiting..." for case 3

Q17 -- Menu Integration

Write a small `main` method that:

- Displays the menu\
- Reads the option\
- Loops until exit\
- Uses your switch from Q16

(You may use pseudocode or Java.)
