

# Self Practice - Introduction to Java

1. Determine whether to sleep in based on two conditions: `weekday` and `vacation` . If it's not a weekday or if we're on vacation, we sleep in. Write Java code to determine whether we sleep in or not.

Input - 1:

```
weekday = false, vacation = false
```

Output - 1:

```
true
```

Input - 2:

```
weekday = true, vacation = false
```

Output - 2:

```
false
```

Input - 3:

```
weekday = false, vacation = true
```

Output - 3:

```
true
```

Analysis:

- We have two conditions: `weekday` and `vacation` .

- If it's not a weekday or if we're on vacation, we sleep in.
- We can use the `if` statement to check the conditions.
- If the weekday is false or if we're on vacation, we sleep in.
- If both conditions are false, we don't sleep in.
- We can write a Java program to determine whether we sleep in or not.

## Java Code:

```
package com.self_practice;

import java.util.Scanner;

public class SleepIn {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        boolean weekday = scanner.nextBoolean();
        boolean vacation = scanner.nextBoolean();
        boolean sleepIn = !weekday || vacation;
        System.out.println(sleepIn);
    }
}
```

## Output:

```
true
false
true
```

---

2. Determine whether we are in trouble based on the smiling status of two monkeys, `aSmile` and `bSmile`. We are in trouble if both monkeys are smiling or if neither of them is smiling. Return true if we are in trouble

Input - 1:

```
aSmile = false, bSmile = false
```

### Output - 1:

```
true
```

### Input - 2:

```
aSmile = true, bSmile = true
```

### Output - 2:

```
true
```

### Input - 3:

```
aSmile = false, bSmile = true
```

### Output - 3:

```
false
```

## Analysis:

- We have two monkeys: `aSmile` and `bSmile` .
- We are in trouble if both monkeys are smiling or if neither of them is smiling.
- We can use the `if` statement to check the conditions.
- If both monkeys are smiling or if neither of them is smiling, we are in trouble.
- If only one monkey is smiling, we are not in trouble.
- We can write a Java program to determine whether we are in trouble or not.

## Java Code:

```
package com.self_practice;
```

```
import java.util.Scanner;

public class MonkeyTrouble {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        boolean aSmile = scanner.nextBoolean();
        boolean bSmile = scanner.nextBoolean();
        boolean inTrouble = aSmile == bSmile;
        System.out.println(inTrouble);
    }
}
```

## Output:

```
true
true
true
```

---

3. Given two integer values, calculate their sum. If the two values are the same, return double their sum; otherwise, return their sum.

### Input-1:

```
a = 1, b = 2
```

### Output-1:

```
3
```

### Input-2:

```
a = 3, b = 2
```

### Output-2:

5

### Input-3:

a = 2, b = 2

### Output-3:

8

## Analysis:

- We have two integer values: `a` and `b`.
- We need to calculate their sum.
- If the two values are the same, return double their sum.
- Otherwise, return their sum.

## Java Code:

```
package com.self_practice;

import java.util.Scanner;

public class SumDouble {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int a = scanner.nextInt();
        int b = scanner.nextInt();
        int sum = a + b;
        int result = a == b ? sum * 2 : sum;
        System.out.println(result);
    }
}
```

## Output:

```
3
5
8
```

---

4. We have a loud-talking parrot. The "hour" parameter is the current hour time in the range 0..23. We are in trouble if the parrot is talking and the hour is before 7 or after 20. Print true if we are in trouble.

**Input-1:**

```
hour = 6, talking = true
```

**Output-1:**

```
true
```

**Input-2:**

```
hour = 7, talking = true
```

**Output-2:**

```
false
```

**Input-3:**

```
hour = 6, talking = false
```

**Output-3:**

```
false
```

## Analysis:

- We have a loud-talking parrot.
- The "hour" parameter is the current hour time in the range 0..23.
- We are in trouble if the parrot is talking and the hour is before 7 or after 20.

## Java Code:

```
package com.self_practice;

import java.util.Scanner;

public class ParrotTrouble {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int hour = scanner.nextInt();
        boolean talking = scanner.nextBoolean();
        boolean inTrouble = talking && (hour < 7 || hour > 20);
        System.out.println(inTrouble);
    }
}
```

## Output:

```
6
true
true
```

---

5. We're hosting a party with tea and candy. The outcome of the party is encoded as follows: 0=bad, 1=good, or 2=great. A party is considered good (1) if both tea and candy are at least 5. If either tea or candy is at least double the amount of the other one, the party is great (2). However, if either tea or candy is less than 5, the party is always bad (0).

Input-1:

```
tea = 6, candy = 8
```

### Output-1:

```
1
```

### Input-2:

```
tea = 3, candy = 8
```

### Output-2:

```
0
```

### Input-3:

```
tea = 20, candy = 6
```

### Output-3:

```
2
```

## Analysis:

- We're hosting a party with tea and candy.
- The outcome of the party is encoded as follows: 0=bad, 1=good, or 2=great.
- A party is considered good (1) if both tea and candy are at least 5.
- If either tea or candy is at least double the amount of the other one, the party is great (2).
- However, if either tea or candy is less than 5, the party is always bad (0).

## Java Code:

```
package com.self_practice;  
  
import java.util.Scanner;
```



```

public class TeaParty {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int tea = scanner.nextInt();
        int candy = scanner.nextInt();
        int result = 0;
        if (tea >= 5 && candy >= 5) {
            result = 1;
            if (tea >= 2 * candy || candy >= 2 * tea) {
                result = 2;
            }
        }
        System.out.println(result);
    }
}

```

## Output:

```

6
8
1

```

**6. Identify the required variables to store the information of a mobile for a mobile shop. Create and show the sample data.**

## Analysis:

- We need to store the information of a mobile for a mobile shop.
- We can use the following variables to store the information:
  - `brand` (String): the brand of the mobile.
  - `model` (String): the model of the mobile.
  - `color` (String): the color of the mobile.
  - `price` (double): the price of the mobile.
  - `ram` (int): the RAM of the mobile.
  - `storage` (int): the storage of the mobile.
  - `camera` (int): the camera resolution of the mobile.

- `battery` (int): the battery capacity of the mobile.
- We can create sample data for the mobile and display it.

## Java Code:

```
package com.self_practice;

public class MobileShop {
    public static void main(String[] args) {
        String brand = "Samsung";
        String model = "Galaxy S21 Ultra";
        String color = "Phantom Black";
        double price = 1199.99;
        int ram = 12;
        int storage = 256;
        int camera = 108;
        int battery = 5000;

        System.out.println("Brand: " + brand);
        System.out.println("Model: " + model);
        System.out.println("Color: " + color);
        System.out.println("Price: Rs. " + price);
        System.out.println("RAM: " + ram + " GB");
        System.out.println("Storage: " + storage + " GB");
        System.out.println("Camera: " + camera + " MP");
        System.out.println("Battery: " + battery + " mAh");
    }
}
```

## Output:

```
Brand: Samsung
Model: Galaxy S21 Ultra
Color: Phantom Black
Price: Rs. 41999.99
RAM: 12 GB
Storage: 256 GB
Camera: 108 MP
Battery: 5000 mAh
```

## 7. Do the following and explore what is happening. (Print the values to see the impact)

- Initialize an integer variable 'price' with the value 52 and assign it to a float variable 'priceInFloat'.
- Initialize a byte variable 'age' with the value 83 and assign it to an integer variable 'intAge'.
- Initialize an integer variable 'highestAge' to 120 and assign it to a byte variable 'byteAge'.
- Change the value of 'highestAge' to 130 and assign it to 'byteAge'.

## Analysis:

- We will explore the impact of initializing variables of different data types and assigning them to other variables of different data types.
- We will see how the values are stored and printed when we assign variables of one data type to variables of another data type.

## Java Code:

```
package com.self_practice;

public class VariableAssignment {
    public static void main(String[] args) {
        // Initialize an integer variable 'price' with the value 52 and
        // assign it to a float variable 'priceInFloat'.
        int price = 52;
        float priceInFloat = price;
        System.out.println("price: " + price);
        System.out.println("priceInFloat: " + priceInFloat);

        // Initialize a byte variable 'age' with the value 83 and
        // assign it to an integer variable 'intAge'.
        byte age = 83;
        int intAge = age;
        System.out.println("age: " + age);
        System.out.println("intAge: " + intAge);

        // Initialize an integer variable 'highestAge' to 120 and
        // assign it to a byte variable 'byteAge'.
        int highestAge = 120;
```

```

byte byteAge = (byte) highestAge;
System.out.println("highestAge: " + highestAge);
System.out.println("byteAge: " + byteAge);

// Change the value of 'highestAge' to 130 and
// assign it to 'byteAge'.
highestAge = 130;
byteAge = (byte) highestAge;
System.out.println("highestAge: " + highestAge);
System.out.println("byteAge: " + byteAge);
}
}

```

## Output:

```

price: 52
priceInFloat: 52.0
age: 83
intAge: 83
highestAge: 120
byteAge: 120
highestAge: 130
byteAge: -126

```

8. If the selling price of 15 items and total profit earned on them is input through the keyboard, write a program to find the cost price of one item.

## Input:

```
sellingPrice = 60, profit = 30
```

## Output:

```
2
```

## Input:

```
sellingPrice = 225, profit = 45
```

## Output:

```
12
```

## Analysis:

- We have the selling price of 15 items and the total profit earned on them.
- We need to find the cost price of one item.
- We can use the formula:  $\text{costPrice} = (\text{sellingPrice} - \text{profit}) / 15$ .
- We can write a Java program to find the cost price of one item.

## Java Code:

```
package com.self_practice;

import java.util.Scanner;

public class CostPrice {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int sellingPrice = scanner.nextInt();
        int profit = scanner.nextInt();
        int costPrice = (sellingPrice - profit) / 15;
        System.out.println(costPrice);
    }
}
```

## Output:

```
60
30
2
```

---

9. Given 2 non-negative ints, a and b, return their sum, so long as the sum has the same number of digits as a. If the sum has more digits than a, just return a without b.

Input:

```
a = 2, b = 3
```

Output:

```
5
```

Input:

```
a = 8, b = 3
```

Output:

```
8
```

## Analysis:

- We have two non-negative integers: `a` and `b`.
- We need to return their sum, so long as the sum has the same number of digits as `a`.
- If the sum has more digits than `a`, just return `a` without `b`.
- We can write a Java program to return the sum based on the given conditions.

## Java Code:

```
package com.self_practice;

import java.util.Scanner;

public class SumDigits {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int a = scanner.nextInt();
```

```

int b = scanner.nextInt();
int sum = a + b;
int sumDigits = String.valueOf(sum).length();
int aDigits = String.valueOf(a).length();
if (sumDigits == aDigits) {
    System.out.println(sum);
} else {
    System.out.println(a);
}
}
}

```

## Output:

```

2
3
5

```

10. Given an int n, return the string form of the number followed by "!". So the int 6 yields "6!". Except if the number is divisible by 3 use "Fizz" instead of the number, and if the number is divisible by 5 use "Buzz", and if divisible by both 3 and 5, use "FizzBuzz". Note: the % "mod" operator computes the remainder after division, so 23 % 10 yields 3. What will the remainder be when one number divides evenly into another?

## Analysis:

- We have an integer `n`.
- We need to return the string form of the number followed by "!".
- If the number is divisible by 3, use "Fizz" instead of the number.
- If the number is divisible by 5, use "Buzz" instead of the number.
- If the number is divisible by both 3 and 5, use "FizzBuzz".
- We can use the `%` operator to check if the number is divisible by another number.
- We can write a Java program to return the string based on the given conditions.

## Java Code:

```
package com.self_practice;

import java.util.Scanner;

public class FizzBuzz {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        if (n % 3 == 0 && n % 5 == 0) {
            System.out.println("FizzBuzz!");
        } else if (n % 3 == 0) {
            System.out.println("Fizz!");
        } else if (n % 5 == 0) {
            System.out.println("Buzz!");
        } else {
            System.out.println(n + "!");
        }
    }
}
```

## Output:

```
6
Fizz!
```

---