JOBSHEET 3

Variables, Data Types, Operators and Input-Output

Name
Sherly Lutfi Azkiah Sulistyawati

NIM
2341720241

Class
1I

Department
Information Technology

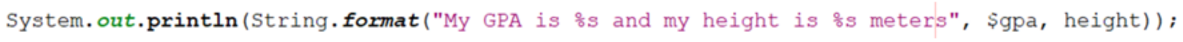
Study Program
D4 Informatics Engineering

Labs Activity

Question! (Experiment 1)

1. Change the variable name so that the variable naming model is good and correct!

2. What is the function of %s in the statement below?

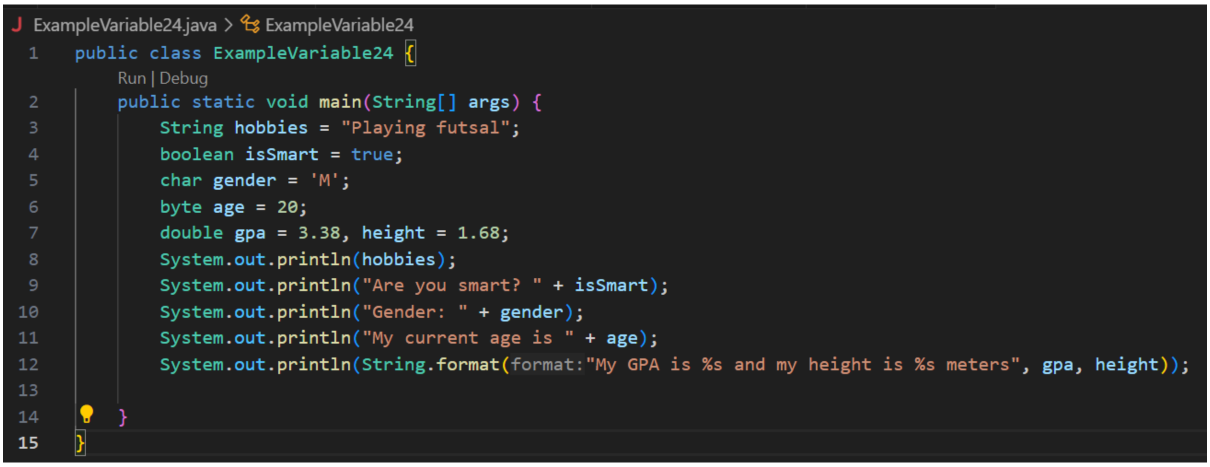


```
System.out.println(String.format("My GPA is %s and my height is %s meters", $gpa, height));
```

Is there anything you can use instead of %s? Name and explain it!

Answer!

1.



```
public class ExampleVariable24 {
    public static void main(String[] args) {
        String hobbies = "Playing futsal";
        boolean isSmart = true;
        char gender = 'M';
        byte age = 20;
        double gpa = 3.38, height = 1.68;
        System.out.println(hobbies);
        System.out.println("Are you smart? " + isSmart);
        System.out.println("Gender: " + gender);
        System.out.println("My current age is " + age);
        System.out.println(String.format(format:"My GPA is %s and my height is %s meters", gpa, height));

    }
}
```

2. In Java's string formatting, %s is used as a format specifier to represent a placeholder for a string. There are other format specifiers available for different data types. For example; %d format specifier is used for formatting integers; %c format specifier is used for formatting characters; %b format specifier is used for formatting boolean values, and so on.

Question! (Experiment 2)

1. Explain why the **bloodGroup** does not display an "A"!

2. Explain the meaning of **byte distance = (byte) 130**! Then, explain why the results change when displayed!

3. In the syntax **float temperature = 60.50F;** remove the letter **F**, then run again. What happened?

4. Why does the result change when displaying weight values?

5. Explain the meaning of initializing **0x10** on **number** variables! What does it do?

Answer!

1. The reason of display is “65” instead of the character 'A' is that explicitly cast bloodGroup to a byte before printing it. If we want to display the character 'A', we should simply print the bloodGroup variable without the cast. And why display “65” because the ASCII value of 'A' is 65.
2. In Java, byte data type can represent value in the range of -127 to 128. If we assign a value outside this range, it would be wrap-around. So when we exceed the maximum positive value (130), the value wraps around to the minimum negative value (-126).
3. It will be error. The letter 'F' at the end of the number '60.50' indicates that the value should be treated as a float literal. In Java, if we don't provide the 'F' or 'f' suffix for floating-point literals, the compiler treats them as double by default.
4. The result of displaying the weight value changes because we casting it to a float before concatenating it with the string and float data type has less precision compared to double.
5. "0x" indicates that the number is in hexadecimal (base 16) format. In this case, 0x10 is equivalent to the decimal number 16. So, the variable number will hold the integer value 16.

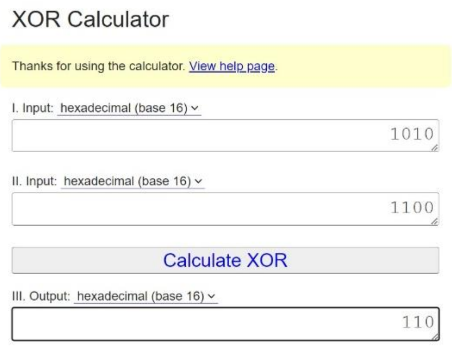
Question! (Experiment 3)

1. Explain in your opinion what is the difference between **x++** and **++x**!
2. What is the result of int **z = x ^ y;** do the calculations manually (you can use a calculator)!

Answer!

1. In x++ the value of variable is printed first then it is incremented whereas in ++x the value is incremented first and then it is displayed.
2. x = 10, y = 12
   x = 1010
   y = 1100

   z = 0110 = 6



XOR Calculator

Thanks for using the calculator. View help page.

I. Input: hexadecimal (base 16)

1010

II. Input: hexadecimal (base 16)

1100

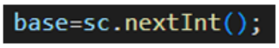
Calculate XOR

III. Output: hexadecimal (base 16)

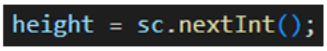
110

Question! (Experiment 4)

1. Explain why you have to declare Scanner in Experiment 4?
2. Explain the use of the program snippets below!



```
base=sc.nextInt();
```



```
height = sc.nextInt();
```

Answer!

1. We have to declaring and initializing the Scanner to read input from the user, and be able to interact with external data sources. Declaring and initializing a Scanner is a fundamental step in many programming tasks that involve user input or file input.
2. The program snippets base = sc.nextInt(); and height = sc.nextInt(); are used to read integer input values from the user or from some input source using a Scanner object in Java.

Assignment

1. Do assignments according to your group's final project topic!
    a. Identify input, output, processes based on the scope of each group's final project topic. The processes identified are limited to processes that use arithmetic operators.

    b. Identify variables and data types based on input, output, and process according to project topic based on 1a.

    c. Implement questions a and b into Java program code so that it becomes a program that utilizes variables, data types, data input, arithmetic processes to display the expected output.

Answer!

1. a. Input = name, category, workHours, salaryPerHour, overtime

    Output = basicSalary, bonus, salary

    Process = input name -> input category -> input workHours -> input salaryPerHour -> input overtime -> calculate basicSalary -> calculate bonus -> calculate salary

    b. String name

    int category

    int workHours

    int salaryPerHour

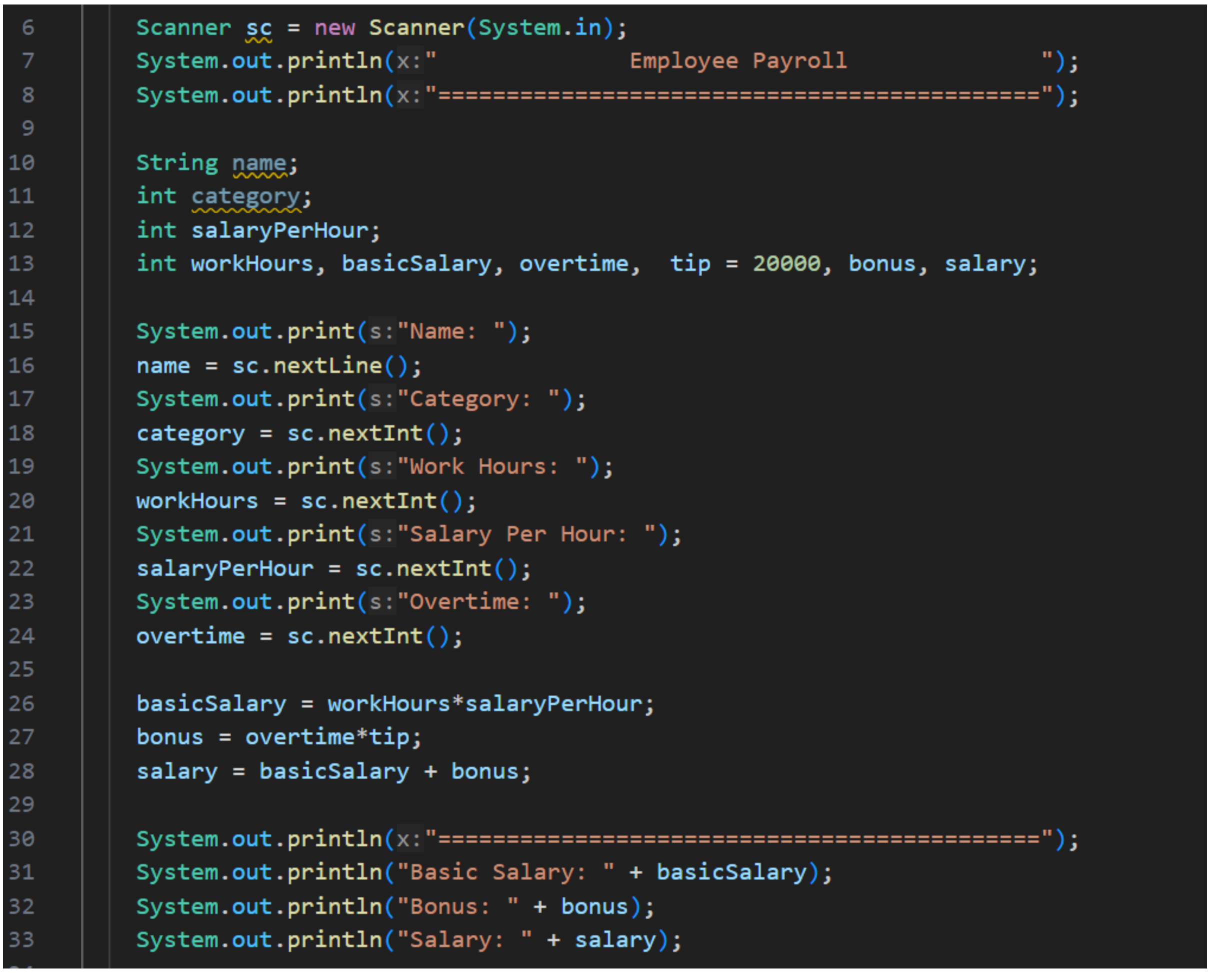
int overtime

int tip = 20000

int basicSalary

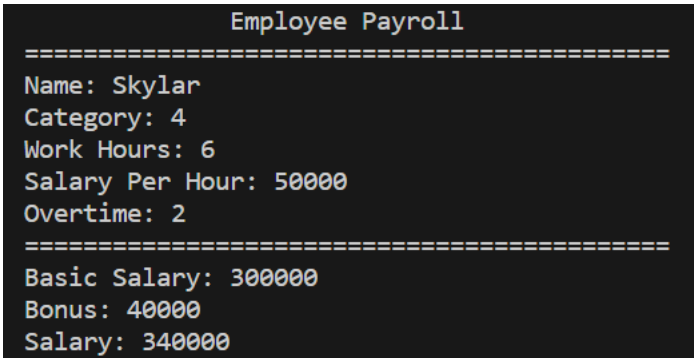
int bonus

int salary



```java
        Scanner sc = new Scanner(System.in);
        System.out.println(x:"              Employee Payroll              ");
        System.out.println(x:"============================================");

        String name;
        int category;
        int salaryPerHour;
        int workHours, basicSalary, overtime,  tip = 20000, bonus, salary;

        System.out.print(s:"Name: ");
        name = sc.nextLine();
        System.out.print(s:"Category: ");
        category = sc.nextInt();
        System.out.print(s:"Work Hours: ");
        workHours = sc.nextInt();
        System.out.print(s:"Salary Per Hour: ");
        salaryPerHour = sc.nextInt();
        System.out.print(s:"Overtime: ");
        overtime = sc.nextInt();

        basicSalary = workHours*salaryPerHour;
        bonus = overtime*tip;
        salary = basicSalary + bonus;

        System.out.println(x:"============================================");
        System.out.println("Basic Salary: " + basicSalary);
        System.out.println("Bonus: " + bonus);
        System.out.println("Salary: " + salary);
```

c.



```
               Employee Payroll
============================================
Name: Skylar
Category: 4
Work Hours: 6
Salary Per Hour: 50000
Overtime: 2
============================================
Basic Salary: 300000
Bonus: 40000
Salary: 340000
```