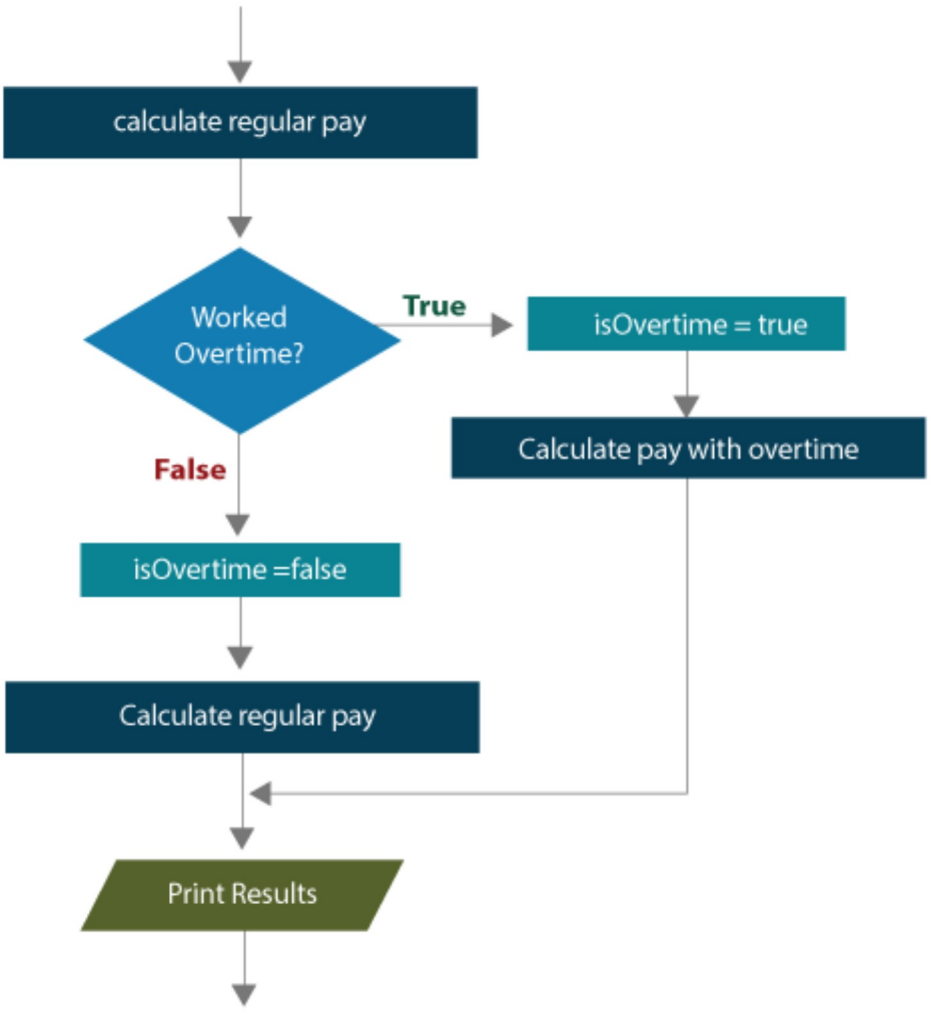
03.04 Virtual Lecture Notes (Part 2)

The SalaryV2 class is a minor variation of the SalaryV1 class. Both programs are designed to calculate an employee's salary based on the hourly pay rate and the number of hours worked. If the employee works more than 40 hours, time-and-a-half is paid for the overtime hours.

- Carefully study the source code for the SalaryV2 class.
- Run the program and observe its performance and output.

Once again, the overall structure of the SalaryV2 class should seem very familiar to you after studying the AdmissionFeeV2 class. Can you spot the double branching conditional statements in the flowchart?



If you compare this to the flowchart for SalaryV1, you will notice a decision point where the flow of control branches based on evaluation of a boolean condition. Compare the flowchart with the corresponding code segment below.

```
…
      if(totalHours > 40)
      {
         isOvertime = true;
         totalSalary = 40 * payRate + (totalHours - 40) *
payRate * 1.5;
      }
      else
       {
         isOvertime = false;
         totalSalary = totalHours * payRate;
      }
      admissionFee += admissionFee * tax;
…
```

How it works:

- Evaluates the **`boolean`** expression within the **`if`** statement. It will be **true** or **false** depending on whether the total hours worked is greater than 40 or not.
- When the **`if`** condition is **true**, the value of **`isOvertime`** is set to **`true`** and the salary, including overtime pay, is calculated. Then the flow of control skips the **`else`** block and continues with the print statements.
- When the **`if`** condition is **`false`**, the code within the **`else`** block is executed. This includes assigning **`false`** to **`isOvertime`** and calculating the pay at the normal rate. The flow of control then continues with the print statements.

Continue to analyze the program line by line and make sure you understand the syntax and purpose of each statement.

Although both versions of this program work equally well, saving two lines of code with the simple if statement does not compensate for its lack of clarity. Until you attain expert status as a programmer, it is better to write code that can be clearly understood, even if it means typing a few extra lines. Consequently, **`if-else`** is a better choice in this situation.

Print