**Lab 5**

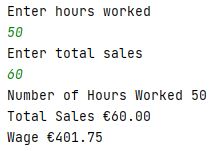
**Exercise 1 (If Statements)**

Employees at Java Burgers earn a basic hourly wage of €7.25. They will receive time-and-a-half (1.5) of their basic rate for overtime hours, i.e. anything over 40 hours. In addition, they will receive a commission on the sales they generate while tending the counter. The commission is based on the following formula:

|  |  |
| --- | --- |
| **Sales Volume** | **Commission** |
| Over €50.00 | 5% of total sales |
| <= 50 | No commission |

Implement the Java program that inputs the number of hours worked and the total sales for the employee and computes and outputs the wage.

**Sample Output**



**Exercise 2 (If else)**

Implement a Java program which allows a user to type in the symbols for logical operators (i.e. &&, ||, !) and outputs a message saying which operator it is (AND, OR, NOT). If the input matches none of these three, output an error message.

**Exercise 3 (If else)**

Implement a Java program that calculates the area of a circle given the radius. The radius measurement should be input by the user. If a negative value is given for the radius then output an error message. Use the Math.PI constant in your program (treat it just like any other constant). Example: area = Math.PI \* (radius \* radius);

Sample Output 1



Sample Output 2



**Exercise 4a (switch statement)**

Implement a Java program that allows the user to input a coinage amount in pence (i.e. 1p, 2p, 5p, 10p, 20p, 50p) and prints a message telling them how much in Euros (cents) that coin is worth. Assume the input is an integer value. If an invalid amount is entered output an error message. Use a switch statement in your solution. £1 = 1.27 EUR.

**Sample Output 1**



**Sample Output 2**



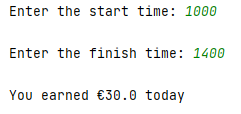
**Exercise 4b**

Rewrite exercise 4a using the new switch statement

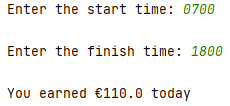
**Exercise 5 (Nested if)**

Implement a Java program that calculates the pay owed to a person for a day's work. They are paid €10 per hour for the first 8 hours they work in a day and time and a half for any hours over that. The user should be prompted to enter the time they started work and the time they finished. All inputs should be in 24-hour format (e.g. 0900, 2100). Assume that the person took 1 hour of breaks during the day for which they aren't to be paid. If the hours worked are less than 0 then print an error message.

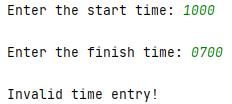
**Sample Output 1**



**Sample Output 2**



**Sample Output 3**



**Exercise 6 (Math.pow)**

Write a loan calculator application that computes both monthly and total payments for a given loan amount, in Euro and cents, annual interest rate, in percent, e.g. 12.5, and loan period, in years e.g. 30. The loan amount, annual interest rate and loan period is to be input by the user.

The formula for computing the monthly payment is as follows:

Monthly payment = ( L\* R) / 1- (1/(1+R))N

The above formula can be written in Java using the Math.pow method:

**Monthly payment = ( L\* R) / 1- math.pow(1+R, -N))**

Where L is the loan amount, R is the monthly interest rate, and N is the number of payments.

The monthly rate R is expressed in a fractional value , e.g. 0.01 for 1 per cent monthly rate. Once the monthly payment is derived the total payment can be determined by multiplying the monthly payment by the number of months the payment is made.

Note that you must convert the annual interest rate to the monthly interest rate and the input loan period to the number of monthly payments. The monthly interest rate is calculated as follows:

Monthly interest rate = annual interest rate / 100.0 / 12

Number of payments = loan period \* 12

where 12 represents the number of months in the year in both cases

**Sample Output**

