The Weekend Hack

FULL STACK WEB DEVELOPMENT 2020

Assigned date: 8th February 2020

Due date: 14th February 2020

Start time: 5:14 pm

End time: 8:14 pm

**Take Home Exam 1**

JavaScript

**JavaScript: Link -** <https://js.do/Dnoel26/js_take_home_exam_1>

<script>

// Edit your script here

//Employee Data Variables---

let employee\_number, employee\_first\_name, employee\_last\_name, employee\_full\_name, hours\_worked, employee\_type\_code, employee\_type;

//Lecturer Only Variables---

let lecturer\_qualification\_code, lecturer\_hourly\_rate, lecturer\_monthly\_allowance;

//Regular Workers---

let fixed\_monthly\_salary, prorated\_monthly\_salary, regular\_worker\_hourly\_rate, monthly\_overtime;

//Payslip Variables---

let income\_tax, health\_surcharge, gross\_monthly\_salary, deductions, net\_monthly\_salary, payslip;

employee\_number = prompt("Please enter a valid employee number");

employee\_first\_name = prompt("Please enter employee's first name");

employee\_last\_name = prompt("Please enter employee's surname");

employee\_full\_name = employee\_first\_name + employee\_last\_name;

employee\_type\_code = prompt("Please enter a valid employee type code \n L for Lecturer \n R for Regular worker");

if(employee\_type\_code == "L" || employee\_type\_code == "R" || employee\_type\_code == "l" || employee\_type\_code == "r")

{

hours\_worked = parseInt(prompt(`Please enter the number of hours worked this month for ${employee\_full\_name}`));

if(employee\_type\_code == "L" || employee\_type\_code == "l")

{

lecturer\_qualification\_code = prompt(`Please enter a valid lecturer qualification code for ${employee\_full\_name} \n B for Bachelor's Degree \n M for Master's Degree`);

employee\_type = "Lecturer";

if(lecturer\_qualification\_code == "M" || lecturer\_qualification\_code == "m")

{

lecturer\_hourly\_rate = 575;

lecturer\_monthly\_allowance = 2500;

gross\_monthly\_salary = (hours\_worked \* lecturer\_hourly\_rate) + lecturer\_monthly\_allowance;

if(gross\_monthly\_salary > 499.99)

{

health\_surcharge = 33.00;

}

else if(gross\_monthly\_salary > 0 && gross\_monthly\_salary <= 499.99)

{

health\_surchage = 19.20;

}

if(gross\_monthly\_salary > 5000)

{

income\_tax = (25 / 100) \* (gross\_monthly\_salary - 5000);

}

else if(gross\_monthly\_salary <= 5000)

{

income\_tax = 0;

}

deductions = health\_surcharge + income\_tax;

net\_monthly\_salary = gross\_monthly\_salary - deductions;

confirm(`Press OK to print pay slip with the following information: \n Employee number: ${employee\_number} \n Employee name: ${employee\_full\_name} \n Employee type: ${employee\_type} \n Employee Gross Salary: ${gross\_monthly\_salary} \n Employee Deductions: ${deductions} \n Employee Net Salary: ${net\_monthly\_salary}`);

}

else if(lecturer\_qualification\_code == "B" || lecturer\_qualification\_code == "b")

{

lecturer\_hourly\_rate = 325;

lecturer\_monthly\_allowance = 1250;

gross\_monthly\_salary = (hours\_worked \* lecturer\_hourly\_rate) + lecturer\_monthly\_allowance;

if(gross\_monthly\_salary > 499.99)

{

health\_surcharge = 33.00;

}

else if(gross\_monthly\_salary > 0 && gross\_monthly\_salary <= 499.99)

{

health\_surcharge = 19.20;

}

if(gross\_monthly\_salary > 5000)

{

income\_tax = (25 / 100) \* (gross\_monthly\_salary - 5000);

}

else if(gross\_monthly\_salary <= 5000)

{

income\_tax = 0;

}

deductions = health\_surcharge + income\_tax;

net\_monthly\_salary = gross\_monthly\_salary - deductions;

confirm(`Press OK to print pay slip with the following information: \n Employee number: ${employee\_number} \n Employee name: ${employee\_full\_name} \n Employee type: ${employee\_type} \n Employee Gross Salary: ${gross\_monthly\_salary} \n Employee Deductions: ${deductions} \n Employee Net Salary: ${net\_monthly\_salary}`);

}

else

{

alert(`Sorry, ${lecturer\_qualification\_code} is not a valid qualification code`);

}

}

if(employee\_type\_code == "R" || employee\_type\_code == "r")

{

fixed\_monthly\_salary = parseFloat(prompt(`Please enter the fixed monthly salary at 160 work hours for ${employee\_full\_name}`)).toFixed(2);

employee\_type = "Regular Worker";

regular\_worker\_hourly\_rate = fixed\_monthly\_salary / 160;

if(hours\_worked >= 0 && hours\_worked < 160)

{

prorated\_monthly\_salary = regular\_worker\_hourly\_rate \* hours\_worked;

gross\_monthly\_salary = prorated\_monthly\_salary;

}

else if(hours\_worked >= 160)

{

overtime = (hours\_worked - 160) \* (regular\_worker\_hourly\_rate) \* 2;

gross\_monthly\_salary = (hours\_worked \* regular\_worker\_hourly\_rate) + overtime;

}

if(gross\_monthly\_salary > 499.99)

{

health\_surcharge = 33.00;

}

else if(gross\_monthly\_salary > 0 && gross\_monthly\_salary < 500)

{

health\_surchage = 19.20;

}

alert(health\_surcharge);

if(gross\_monthly\_salary > 5000)

{

income\_tax = (25 / 100) \* (gross\_monthly\_salary - 5000);

}

else if(gross\_monthly\_salary <= 5000)

{

income\_tax = 0;

}

alert(income\_tax);

deductions = health\_surcharge + income\_tax;

net\_monthly\_salary = gross\_monthly\_salary - deductions;

confirm(`Press OK to print pay slip with the following information: \n Employee number: ${employee\_number} \n Employee name: ${employee\_full\_name} \n Employee type: ${employee\_type} \n Employee Gross Salary: ${gross\_monthly\_salary} \n Employee Deductions: ${deductions} \n Employee Net Salary: ${net\_monthly\_salary}`);

}

}

else

alert(`Sorry, employee type code ${employee\_type\_code} is invalid. Program terminated.`);

</script>

<!-- edit your html here -->