**CSI104: Foundations Of Computer Science**

**Duration: 90’**

**Lab 2: Algorithm**

**Objective**:

* **Define an algorithm and relate it to problem-solving**
* **Define three constructs and describe their use in algorithms.**
* **Describe pseudocode and how they are used in algorithms.**
* **Describe UML diagrams and how they are used in algorithms.**

**Materials:**

Based on exercises of chapter 8 in the textbook “Foundations Of Computer Science, 4nd Edition, Behrouz Forouzan, 2017.”

**Student's task:**

* Review the whole chapter 8 content in the textbook.
* Write down solutions to exercises (step by step)
* Finish exercises and submit the results to the lecturer in class.

**Scoring scale: 10**

Students will represent the algorithm of the problem in two ways: pseudo-code and UML

**Sample: Write an algorithm to calculate the sum of two integers**

Using pseudo-code:

Input: num1, num2  
Processing: result=num1+num2  
Output: print out result

Using UML

Input num1, num2

Result=num1+num2

Print out result

**Exercise 1 (2 marks)**: Write an algorithm to allows users to input two integers and an operator of four operators +, -, \*, / then print out the result to the monitor.

Pseudocode:

Input: num1, num2

Processing:

sum=num1+num2;

sub=num1-num2;

pro=num1\*num2;

div=num1/num2;

Output: print out : sum, sub, pro, div

UML:

Diagram

Description automatically generated

**Exercise 2 (2 marks):** Write an algorithm that will find the smallest integer among five integers: 12, 34, 9, 24, 39

Pseudocode:

Input: 12, 34, 9, 24, 39

Processing:

Min=12

If(min>34){

min=34

}

If (min>9){

min=9

}

If (min>24){

min=24

}

else {

min=39

}

Output:print out min

UML:

Diagram

Description automatically generated

**Exercise 3 (2 marks):** Write an algorithm that will print out the **sum** of **integers** inputted from the keyboard until the value 0 is inputted.

Pseudocode:

Input:Enter intergers n;

Processing:

Sum=0;

For ( ; ; )

If n!= 0 ; continue

If n ==0; sum =i sum +n; break

Output: Print out sum

UML:

Diagram

Description automatically generated

**Exercise 4 (4 marks):**Suppose that:   
In Viet Nam, each people has to pay for his/her yearly personal income tax as the following description:

**Rules:**

***Tax-free income:***

Personal pending amount (tiền nuôi bản thân) **pa=** 9,000,000 d /month

Alimony (tiền cấp dưỡng) for each his/her dependent **pd=** 3,600,000 d/month/dependent

With **n** dependents, Yearly tax-free income: **tf = 12\*(pa + n\*pd)**

***Taxable income (thu nhập chịu thuế)***

**ti = income – tf**

**( If ti<=0 then income tax = 0)**

**Based on taxable income, the employee has to pay his/her income tax with levels pre-defined in the following table:**

|  |  |  |
| --- | --- | --- |
| ***Level*** | ***Taxable Income*** | ***Income tax*** |
| 1 | Less than or equal to 5.000.000 | 5% |
| 2 | From 5.000.001 to 10.000.000 | 10% |
| 3 | From 10.000.001 to 18.000.000 | 15% |
| 4 | Over 18.000.000 | 20% |

Write an algorithm that will calculate and print out : income, ti (Taxable Income) and income tax.

Input: n, income

Processing:

pa = 9 000 000

pd = 3 600 000

tf=12\*(pa +n\*pd)

ti= income – tf

if (ti <= 0 ) {

income tax = 0

}

else if (ti > 0 && ti <= 5 000 000 ) {

income tax = ti \* 5/100;

}

else if (ti >= 5 000 001 && ti <= 10 000 000) {

income tax = 5 000 000 \* 5/100 + (ti – 5 000 000) \* 10/100

}

else if (ti >= 10 000 001 && ti <= 18 000 000) {

income tax = 5 000 000 \* 5/100 + 5 000 000 \* 10/100 + (ti – 10 000 000) \* 15/100

}

else {

income tax = 5 000 000 \* 5/100 + 5 000 000 \* 10/100 + 8 000 000 \* 15/100 + (ti –

18 000 000) \* 20/100

}

return 0

Output: print out income, ti, income tax