

This revision sheets contains point-wise discussions that were held in the class. Therefore merely reading this document may NOT be much helpful and you HAVE TO recall the examples and classroom's whiteboard content along with these points.

Algorithms are nothing new in our life. We have been accomplishing lot of our targets following some algorithm.

Some of those were discussed in the class

- Tyre changing when its gets flat while driving - coming to university for this class - celebrating your admissions here with your family and friends

Then we discussed another example of developing a Google/Bing like search Engine

- Just an overview/sketch of the development modules need to be explained if developer has already taken my final year course of NLP and Computational Linear Algebra. (told the main modules and name of algorithms) - One year junior students may also get the task done but they need relatively detailed instruction about the tasks to be performed since they might not know about Vector space model and related query-to-document matching algorithms. - This tasks may also be get done by those who have completed just one year of their BS studies but they will need details of every step to be performed.

Another example discussed was the need to automatically group a large number of electronic documents using computers - these documents may be news items to be automatically routed to the

sub-editor of different sections of a newspaper -or may be reports collected by security agencies that need to be grouped based on the events discussed in those documents, or health related articles to be grouped according to suitability for readers who may be doctors, patients, researchers/scientists, public awareness etc. (an article suitable for public awareness may be useless for a doctor who might already know it)

Page 1 of

7

In Summary - I can get this done by any level of developer. For every level of developer I need to provide him/her sufficient level of details according to their level of understanding.

Similarly, when you have to get your

To start with, we will develop a general understanding of tasks processed by computers, you need HLL instead of using any specific like C/C++, Python, to comprehend the level of understanding that a computer possesses. Actually we use, most of the time, some “high level language” to instruct a computer what to do for us, so we need to understand a “high level” language (HLL).

PHP, Visual Basic or any other. This will make future learning of any new programming language easier. Due to this reason we will learn an Algorithmic language that resembles a high level programming language. It is designed with common features that are found in most of the programming languages.

We all can add two numbers that were given by someone else that means we

After few weeks we will start using Python to implement your solution steps. I earlier batches C# had been used for years. But these language are just “tools” to get your work know the algorithm to add two numbers.

done. Actual thing to learn is the thought process from start Right? {You were then asked to write this to end for solving problems using computers; analogous to algorithm.}

brushes and colors that every artist uses but a “classic” piece of art starts with an idea and its sketch that an artist bears in mind. Choice of oil paints or water color or Acrylic paints, Pastels etc. comes second. A person with all types of paints but without a theme and its sketch in mind can’t Algorithms discussed in starting few

create even an ordinary painting. weeks:

Add two user-supplied numbers

Multiply two user-supplied numbers

Subtract two user-supplied numbers – there were at least three versions of this algorithm.

Moreover

- there was ambiguity in exact understanding of the algorithmic instructions - therefore we switched to pseudo code instead of algorithm and introduced, Input, Output and calculation instructions.

The pseudo code for an “Ask” type statement was “Input” followed by a variable name – example “Input a” or “Input radius” ; anything that is written after “input” is just a label to hold the value that was asked from the user a computer does not tries to understand that label. For example if you write “Input Angle-in-Radians” computer can’t distinguish between radians and degrees and accepts the user supplied value “as is” and stores this values as “Angle-in-Radians” or in other words it creates a variable named “Angle-in-Radians” and assigns the user supplied value to this variable.

To avoid some confusions that usually arises at beginner level, we have put some additional restrictions in pseudo codes that are usually allowed in programming languages but these restrictions are relaxed

Page 2 of 7

depending upon the progress of the class. One such restriction is to take input of one variable only in an input statement although most of the programming language allow input of more than one variables in an input statement.

Similarly “Tell” type statement is represented in pseudo code by “Output” statement. Syntax of output statement is “Output <Variable-Name>” (at this point, recall the syntax conventions discussed in the class). This <Variable- Name> must be the one used in the input statement or variable that was used to store some value calculated by a calculation statement (similar to algebraic expressions; example  $c=a+b$ )

For calculation statements, initially allowed in the class, mathematical statements were +, -, /, \* and ^ used for addition, subtraction, division, multiplication and exponent (power) respectively.

Order of evaluation of these operators is not same as of mathematical BODMAS rule. In programming languages operators have precedence/priority values. Mathematical (as well as logical) expressions that appear in a program are evaluated in various steps, one operator is evaluated in each step and order of evaluation of these operators are based on its priority. An operator with highest priority amongst those that are present in the expression under process, is evaluated first. Then operator with next higher priority. If more than one operators are of same precedence than leftmost is evaluated earlier.

[Some in-class exercises & Assignments are given here for your practice again](#)

### 1. Divide the two user-supplied numbers

How do computers execute HLL:

Computer does not understands HLL. The only language that it understands is Machine Language therefore a HLL must be translated to Machine language so that computer can act according to our instructions using these translated-into-machine- language instructions. The actual software used to translate depends on HLL but the category of such type of softwares (that translate HLL into Machine language) is called “Compilers”.

There are some other ways too, for example, there are utilities called “Interpreters” that go on reading HLL instructions one by one, translate each it into Machine language and execute it, but some of them also store these translated instructions into a file, called “Executable File”, so that this translation process need not to be performed again to execute those HLL instructions.

Now-a-days there are far more ways to execute HLL instructions. One of these is to translate these HLL instructions into some language, (for example Microsoft's defined MSIL (Microsoft Intermediate Language) for its dot-net

(.NET) environment), that is not Machine language but there are utilities that make suitable arrangement to get these instructions executed by the computers. These utilities may be integrated with the operating system (OS); for example “.NET run time” is integrated MS Windows that gets makes these .NET instructions executed by the computer with the help of OS.

Another such example is of Java language whose compiler produces “Byte Code” that is executed by Java Virtual Machine (JVM).

Assignment questions:

1. Compiler and interpreters did their job of making it possible for computers to run HLL instructions for a long time. Then why JVM and .Net environments were introduced as an alternate? 2. What is the latest version of Microsoft .NET and its distinguishing features? 3. What is latest version of Java / JVM? Which organization / standardization body makes these standards?

Page 3 of 7

2. Take input the radius of a circle and print its area =  $\pi r^2$  and circumference =  $2 \pi r$  3. Take input the radius of an sphere and print its area =  $\frac{4}{3} \pi r^3$  4. Take input cost of per square foot price of a carpet and width and depth of a room in feet then print cost of wall to wall carpet 5. Ask selling price, discount in percentage and print discount price. 6. Take temperature in centigrade and print its equivalent Fahrenheit.

### Conditional execution of some a programming statement

Then we discussed if statement to allow or deny execution of a programming statement. The syntax of if-statement is (RECALL the syntax writing rules discussed in the class specially the square brackets and less-than and greater than signs

if –statement ::= if ( <condition> ) <statement>

and syntax for condition may be one of the following

1. <variable> <operator> <value> 2. <variable>  
<operator> <variable> 3. <value> <operator>  
<variable> 4. <value> <operator> <value>

The fourth one, which is valid according to syntax, is useless since comparing two value always have definite answer, i.e. if it is true it will always be true hence the statements will execute every time and if it is false it will result in always skipping the associated statements.

Operator may be one of the six comparison operators

> (Greater than) < (Less than) >=  
(Greater than or equals to) <=

(Less than or equals to) != (Not equals to) == (Equals to)

When a statement containing a condition is executed it checks whether the value on left and value on right has the stated relationship, or in other words, it checks whether the statement written as condition is true or false. For example, the condition “age >= 18”, results true if value of the variable named “age” is exactly 18 or more then 18.

This is useful in the cases when some conditions are to be met before performing some operations. For example, making sure about a value being positive before calculating its square root, since real-numbered-square-root of negative number does not exist mathematically. Or making sure that denominator is not zero, before division. Similarly, while printing voter’s list from a list of citizens, making sure that a person has attained the minimum age requirement for printing voter list. These are few cases to kick start the thinking of “if” statement’s usage.

Page 4 of  
7

### Changing the sequence of execution:

Steps of an algorithms are executed in sequence in which these are presented to a machine, i.e. execution starts from the first statement then second, third and so on until last statement. This sequence sometime need to be altered for which there is a “goto” statement. Syntax is (remember that algorithmic statements are not case-sensitive)

Goto <Line-number> | <label>

When a “goto” statement on any line of an algorithm executes, then next statement to be executed is set to be the one on <Line-Number> and normal processing is resumed from that line onwards. This explanation will be easier to understand with the help of following example;

Suppose there is a 10 line algorithm without a goto statement (the symbol “...” represents any of the algorithmic statement discussed so far, except the goto statement);

1. ...
2. ...
3. ...
4. ...

5. ...
6. ...
7. ...
8. ...
9. ...
10. ...

Sequence of execution of these statements will be 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10. But with a goto statement as shown below; the sequence of execution will be 1, 2, 3, 4, 9, and 10.

1. ... 2. ...
3. ... 4.
- Goto 9 5.
- ... 6. ... 7.
- ... 8. ... 9.
- ... 10. ...

And sequence of following statements will be an infinite sequence of statements 1, 2, 3, 4, 5, 6, 7, 3, 4, 5, 6, 7, 3, 4, 5, 6, 7, 3, 4, ...

1. ...
2. ...
3. ...
4. ...

Page 5 of  
7

5. ... 6. ...
7. Goto 3
8. ... 9. ...
10. ...

Like all other algorithmic statements, the goto statement may also be preceded by an if statement and in that case goto will be executed only when the condition of the if-statement is true.

1 – input o //o = obtained marks 2 – if (o < 0) goto 1 // since obtained marks can't be negative 3 – input m //m = maximum marks 4 – if (m < 0) goto 3 // since max. marks can't be negative 5 – if

`(o > m) goto 1` 6 – `p = o / m * 100` // `p`= percentage 7 – output `p`

The above algorithm accepts a value for obtained marks and on the next line checks if it is negative; in such case it goes to step 1 and from that point starts the execution normally ("normally" means execute all steps of algorithm starting from step 1 to onwards) all statements from step 1 to onward as it normally does.

The above algorithm accepts a value for obtained marks and on the next line checks if it is negative; if the condition is fulfilled, it then executes the goto command which takes the execution to line 1 (as specified in the goto command). All statements from line 1 and onwards are then executed.

While doing so it takes input and then comes to step 2 where it again checks the user entered value for obtained marks (stored internally with name of "o") according to the condition written beside the "if", if the condition is false then "goto 1" is not executed. If statement is executed whenever flow of algorithm reaches to this step; but statement associated with if is executed only when condition is true but if the condition is false the statement beside the condition is not executed and statement on the line is considered as executed.

When the execution reaches line 4 it checks if maximum marks are negative; since maximum marks cannot be negative, the goto command is used to go back to line 3 where maximum marks are entered again. Execution continues and if statement on line 5 is executed whenever flow of algorithm reaches to this step. Whenever the if condition is fulfilled (i.e the condition statement written is TRUE) only then the associated statement is executed (in this case it is the goto statement) otherwise (i.e the condition is not fulfilled) lines after the if clause are executed.

TRY THESE TWO QUESTIONS FIRST TO JUDGE YOURSELF:

Question 1: Agriculture fields in Pakistan are measured in acres. One acre is of 43,560 square feet.

Write an algorithm/pseudo code that takes input of length and width of a farmer's field in yards from the user and displays area of the field in acres.

Question 2: Government announced public holidays for this year are available at

<https://www.interior.gov.pk/index.php/downloads/category/3-public-notices?download=190:public-holidays-for-2020>

Write a program that takes input a date as dd/mm/yyyy and prints it as dd-MonthName-yyy. If the date is one of those 10 that are announced as public holidays then your program should also print name of that occasion.

IF YOU FIND DIFFICULT TO WRITE PSEUDOCODE OF ABOVE TWO PROBLEMT THEN YOU MUST SOLVE THESE PROBLEMS IN THE SEQUENCE MENTIONED BELOW;

OTHERS SHOULD ALSO DO THESE EXERCISES TO SPEEDUP THEIR ALGORITHMIC THINKING PROCESS.

Available  
at

<http://link.springer.com/openurl?genre=book&isbn=978-3-319-14240-1>

Exercise Numbers are given below (variants of many of these are covered as example/exercise during the class room lectures)

BASICS → 5, 6, 7, 8, 10, 11, 14, 24, 29,  
30

ARITHMATIC → 12, 16, 17, 18, 19, 20, 21, 22, 23, 25, 27,  
28

IF → 32, 34, 36, 37, 38, 39, 40, 45, 49, 50, 54, 55, 57,  
58,

EXERCISE QUESTIONS:





