https://lh6.googleusercontent.com/QH_XpzeT48PQZvynU_XD-kWM-YqlR3ruASon0eJU5AL25StyxzTPhITjrUbAOxAl9YEXJ4FdCyPi9Mq-aQoOtDQMOOVwJI9uKdIFuSDD_BlEoEheLEU9rpseMzbRUoCel1PwJnQI=s0**Department of Computer Science**

**University of Engineering and Technology, Lahore**

**Flow charts, logic building, algorithms**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Learning Outcomes:**

Understand and implement flow chart diagrams for any given problem.

Write an algorithm for any given problem.

Develop problem solving skills

**What is an algorithm?**

Algorithm is a finite set of steps which are used to solve a problem.

*Algorithms are not program code*; it is actually a way to describe the steps you will be following to solve any problem, usually written in pseudocode.

First, let us learn how to write an algorithm.

For example, if you are asked to **download a book online**, your **possible steps may be**:

Step 1: Turn on laptop

Step 2: Open Google Chrome

Step 3: Type book name in search bar

Step 4: Find appropriate website to download the book

Step 5: Click Download button

In the same manner, we can write algorithms for any problem. Let us see some more examples.

**Example 1:** Write a program to add two numbers and their result.



**Algorithm:** 

Step 1: Start

Step 2: Declare variables var1, var2 and result

Step 3: Load values in var1 and var2

Step 4: Add var1 and var2 and assign the sum to result

Step 5: print result

Step 6: Stop

**Example 2:** Write a program to take two number from user and sum them

**Algorithm:** 

Step 1: Start

Step 2: print “Enter first number”

Step 3: input number1

Step 4: print “Enter 2nd number”

Step 5: input number2

Step 6: sum = number1 + number2

Step 7: print sum

Step 8: Stop

**Example 3:** Write a program that convert USD to PKR

| **Style 1** | **Style 2** |
| --- | --- |
| Step 1: Start  Step 2: print “Enter value in usd”  Step 3: input usd  Step 4: pkr = usd \* 170  Step 5: print pkr  Step 6: Stop | Step 1: Start  Step 2: display “Enter value in usd”  Step 3: take value from user into usd  Step 4: pkr = usd \* 170  Step 5: display pkr  Step 6: Stop |
| The purpose of the algorithm is to help reader or programmer to write the code so it should be very clear | |

**Example 4:** Write a program to take two values from the user and print the greater number.

Step 1: Start

Step 2: input number1

Step 3: input number2

Step 4: if number1 is greater than number2

Print “number1 is greater”

else

print “number2 is greater”

Step 5: Stop

**Example 5:** Write a program to take the number from the user if number is positive display, you win if number is negative display, you lose if number is zero display match tie. At the end the display game ended.

Step 1: Start

Step 2: print “Enter a number”

Step 3: input num

Step 4: if number is greater than zero

print “You Won”

else if number is less than zero

print “You Lost”

else

print “Match Tie”

Step 5: print “Game Ended”

Step 6: Stop

**Try Yourself:**

**Question 1:** Write a program that takes 5 subject marks as input and display the total marks and percentage. Hint: Every subject has a total of 100 marks.

**Question 2:** Write a C++ program to check whether a triangle is equilateral, scalene, or isosceles: **Isosceles triangle:** In geometry, an isosceles triangle is a triangle that has two sides of equal length. **Equilateral triangle:** In geometry, an equilateral triangle is a triangle in which all three sides are equal. **Scalene triangle:** A scalene triangle is a triangle that has three unequal sides

**Flowcharts:**

Flowcharts are used to represent the flow of an algorithm.

Through this representation, it becomes easier to understand the algorithm. We use different boxes to show the processing and the boxes are connected using arrows, to show the flow of the process. Let us identify the shapes first.

1. **The oval**

Oval is used to represent “**Start**” or “**end**” of the algorithm.



1. **The rectangle**

A step is represented by a **rectangle**. In this shape, you write all the processing of your algorithm.

1. **The arrow**

To indicate the **flow of steps,** arrows are used.



1. **The diamond**

Whenever you have to make a **decision** in your algorithm, you write that **condition** inside the diamond shape.

**5- The parallelogram** 

To represent the input and output, the parallelogram is used.



Now let us try to make a flowchart of previously mentioned examples by using the shapes shown above.

**Example 1:** Write a program to add two numbers and their result.













**Example 4:** Write a program to take two values from the user and print the greater number.



yes no  
   
  
  
  
  
  


***Practice, by creating flowcharts of question1 and question2.***

**Program Structure in Python:**

So far, we have been discussing the algorithms and their representations. Now let us move one step forward to building our program in Python. We will see the basic structure, and then we will shift our algorithms into python codes.

| Algorithm | Processing |
| --- | --- |
| **Example 1:** Write a program to add two numbers and their result.  Step 1: Start  Step 2: Declare variables var1, var2 and result  Step 3: Load values in var1 and var2  Step 4: Add var1 and var2 and assign the sum to result  Step 5: print result  Step 6: Stop | we don’t have to declare variables in python    Output |
| **Example 3:** Write a program that convert USD to PKR  Step 1: Start  Step 2: print “Enter value in usd”  Step 3: input usd  Step 4: pkr = usd \* 170  Step 5: print pkr  Step 6: Stop |  |
| Step 1: Start  Step 2: print “Enter value in usd”  Step 3: input usd  Step 4: pkr = usd \* 170  Step 5: print pkr  Step 6: Stop |  |
| Step 1: Start  Step 2: print “Enter value in usd”  Step 3: input usd  Step 4: pkr = usd \* 170  Step 5: print pkr  Step 6: Stop |  |

**Example 2:** Write a program to take two number from user and sum them

num1=input(“Enter first number”)

num2=input(“Enter 2nd number”)

sum=num1+num2

print(“Sum is: ”,sum)

**Try Yourself:**

**Question 1:** Write a program that takes 5 subject marks as input and display the total marks and percentage. Hint: Every subject has a total of 100 marks.

**Question 2:** Write a C++ program to check whether a triangle is equilateral, scalene, or isosceles: **Isosceles triangle:** In geometry, an isosceles triangle is a triangle that has two sides of equal length. **Equilateral triangle:** In geometry, an equilateral triangle is a triangle in which all three sides are equal. **Scalene triangle:** A scalene triangle is a triangle that has three unequal sides

**Solution:**

**Question 1**

sub1=int(input("Enter marks of 1st subject out of 100"))

sub2=int(input("Enter marks of 2nd subject out of 100"))

sub3=int(input("Enter marks of 3rd subject out of 100"))

sub4=int(input("Enter marks of 4th subject out of 100"))

sub5=int(input("Enter marks of 5th subject out of 100"))

obtained=sub1+sub2+sub3+sub4+sub5

print("Total Marks= ", obtained)

percentage=obtained/500\*100

print("Your Percentage is ", percentage)

**Question 2:**

side1=int(input("Enter length of one side:x "))

side2=int(input("Enter length of 2nd side:y "))

side3=int(input("Enter length of 3rd side:z "))

if side1==side2 and side2==side3:

print("The Triangle is Equilateral")

elif side1==side2 or side2==side3 or side3==side1:

print("The Triangle is Isosceles")

elif not side1==side2 and not side2==side3:

print("The Triangle is Scalene")