Lab 09: Introduction to Flowgorithm

Key Points:

Graphical Flowchart

Objective:

To gain knowledge about flowgorthim and programming language that is based on simple graphical flowcharts

CLO's:

CLO:

Rubrics for Lab

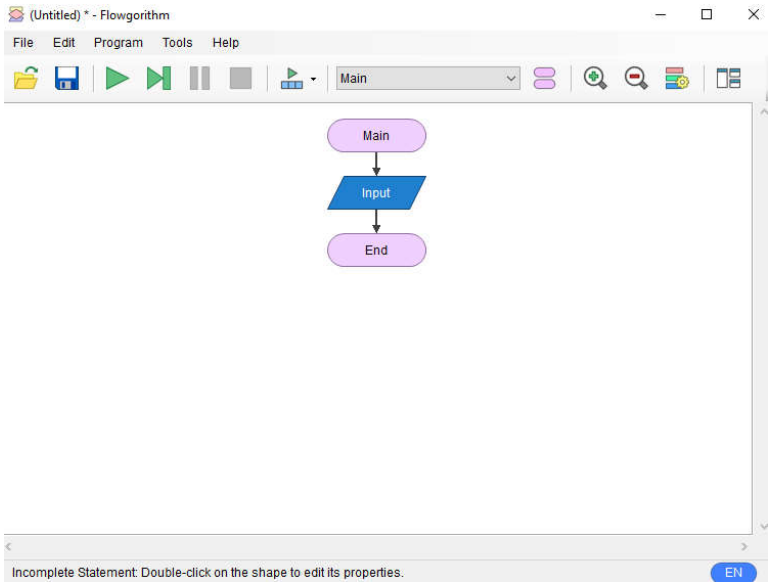
Task	0	1	2	3	4
Making flowcharts on flowgorithm	Student not able to understand the tool	Student able to use basic blocks in making of flowcharts (main, input, output etc.)	Student able to make correct logic	Student able to make advance features (loops, if, else if, functions)	Output is according to the given query

Introduction:

Typically, when you first learns to program, you often use one of the text-based programming languages. Depending on the language, this can either be easy or frustratingly difficult. Many languages require you to write lines of confusing code just to display the text "Hello, world!". By using flowcharts, you can concentrate on programming concepts rather than all the nuances of a typical programming language. You can also run your programs directly in Flowgorithm. Flowchart is a type of diagram that presents a solution to a problem as a broken-down process that you can follow. In other words, this allows you to look at what you did step by step and see what you did wrong or plan out what you want to do. That is the beauty of flowgorithm that scratch, nor any other program modeled after it can give you. The whole idea is that you are learning how to create a flowchart with shapes that can later be manipulated for usage instead of trying to learn complicated terms and functions.

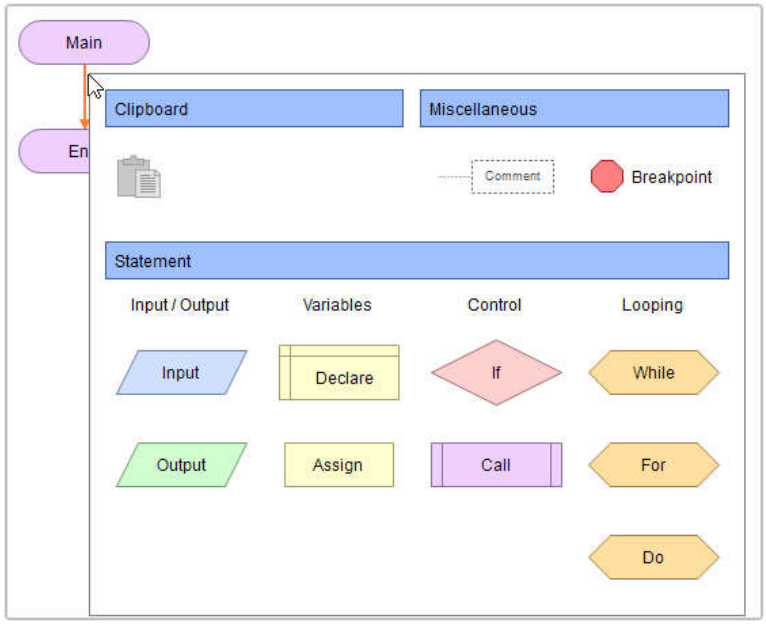
Flowgorithm is useful in many different ways, especially in terms of education. One reason is because people creating simple programs will be able to see their programs change into an actual programming language with the source code viewer. Another reason on top of that is that you can take the converted code, and import it onto a robot, which would make it easier for people to visualize code. Sure you might say that the idea of whether flowgorithim or scratch is easier to use is debatable, but flowgorithm is just as easy for anybody to use as scratch is. The code is displayed in simple shapes that any person of any age would be able to understand. Even some great computer engineers use it to organize the messy codes. And as long as you have installed Windows on your computer, anybody (including people with little or no experience), can use this to transform and create a new program. So I hope that all of you

will find each of these tutorials informative and that this fairly new application will be used widely as an alternative to blocks.
Everything in a flowchart is represented by a shape. You will add your own shapes between the Main and End terminals.



Graphical User Interface

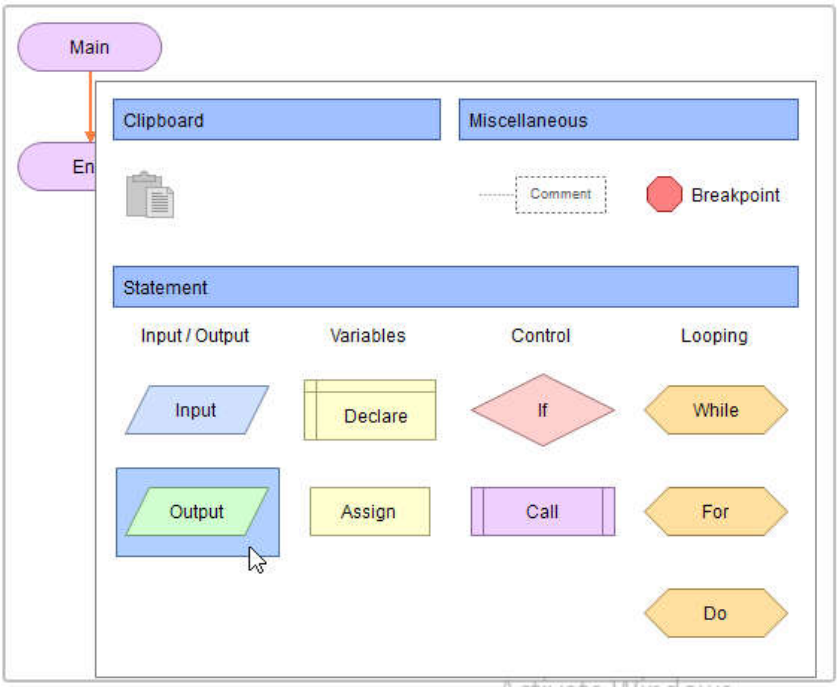
To add a shape, move your mouse pointer over a line. If you can add a shape, the line will turn orange. Now, either double-click or right-click to add a shape. A pop-up menu will appear that shows all the shapes you can add. In flowcharts, each action your computer can perform is represented by a different shape. For example, input and output shapes are represented with parallelograms (slanted rectangles).



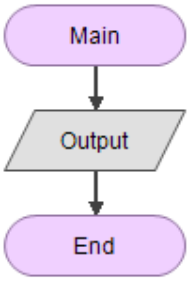
Hello World program:

The Hello World program needs an output shape. It will allow the program to display

information on the screen. Move the mouse over the green parallelogram with the text "Output". It will highlight in blue.
Click on the shape. It will be added to your flowchart.

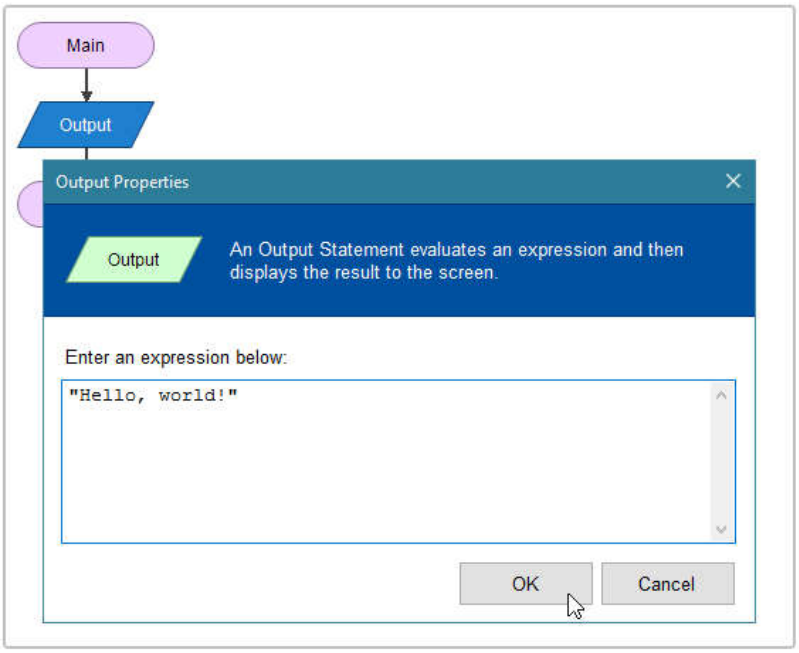


Now your flowchart contains an output shape. Currently, it is displayed in gray. In Flowgorithm, any shape that is gray is "incomplete". This means that the shape needs some information before it can work. You need to tell Flowgorithm what you want it to output. This can be anything from the result of a calculation to a text message. Now, double-click or right-click on the output shape to edit it.



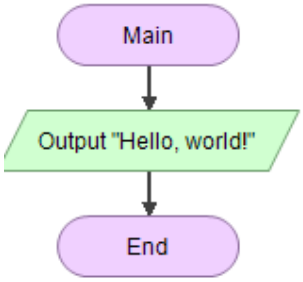
A window will appear titled "Output Properties". Using this window, you can specify what will be displayed on the screen. Output shapes are simple. They will output the result of an expression. This can be something like a text message, a variable, or the result of a calculation.

In the box, type "Hello, world!". You have to add the double-quotes around the text. In computer programming, this is called a String. Now, click on OK to continue. You might get an error window if you didn't type it in correctly. If that happens, just edit the shape again and fix it.

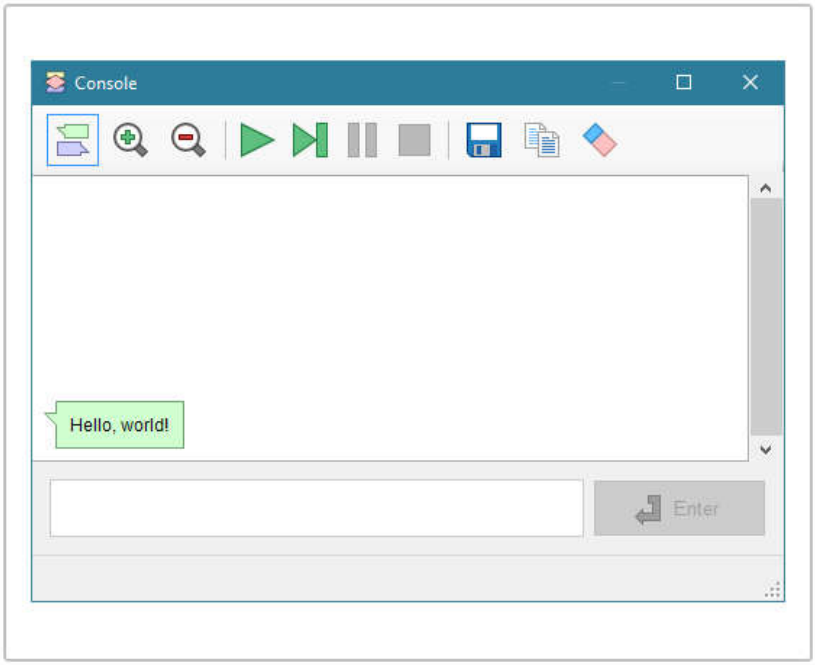


The Output shape will now appear in green. Since all the shapes are in color (nothing is gray), your program is complete. Now it is time to run it. On the main toolbar, you should see a green icon that looks like the Play Button you use when playing a DVD or watching a video on YouTube.

Click on this icon to run your program. You can also press F5 on your keyboard or select Run from the Program menu.



A new window should appear called the "Console Screen". In computer programming, the console is used to display and input textual information. The Console Screen in Flowgorithm was modeled after the texting windows you use on your phone. So, it is like you are texting Flowgorithm!



Data Types:

1. **Integer Data Type:**

The Integer data type is one of the most commonly used types in programming. An integer can store a positive or negative whole number, but can't store fractional values. So, it can store values such as 5, 42, 1947, but can't store numbers such as 3.2, 4.5, etc. If a number with a fractional value is stored into a integer, the fractional value will be discarded. Hence, if 3.2 is stored into an integer, it will only retain 3. The Variable Watch Window displays integers in blue.

2. **Real Data Type**

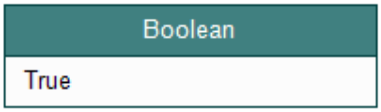
The Real data type can store any number - both whole numbers and ones with fractional values. In many languages, this is called a "double" after the implementation standard known as "double-precision floating point". The Variable Watch Window displays reals in purple.

3. **String Data Type**

The String data type is used to store any textual data. This includes words, letters, or anything else you would send in a text message. In programming, the text is delimited with double quotes. For example: "CSU, Sacramento", "computer", and "Year 1947" are all strings. The Variable Watch Window displays strings in red.

4. **Boolean Data Type**

The Boolean Data Type can store either "true" or "false". These are the basis of decision making in a computer program. The Variable Watch Window displays Booleans in teal.



Naming Rules

Any time you define a function or variable, it is given a unique name called an "identifier". To prevent identifiers from being confused with other items in an expression, they must follow a naming convention. Every programming language has one and they are fairly consistent from language to language.

In Flowgorithm, identifiers must adhere to the following rules:

- They must start with a letter.
- After the first letter, the identifier can contain additional letters or numbers.
- Spaces are not allowed.
- They cannot be key words or words already defined by Flowgorithm

Keywords

Many words using in programming languages overlap with the naming convention used by identifiers. In these cases, the word is "reserved" and cannot be used for Identifiers. In addition, many programming languages predefine functions and other constants. These also cannot be used.

Reserved Words

Flowgorithm only has a few reserved words that are used in expressions.

and	not	true
false	or	
mod	pi	

Data Type Keywords

To prevent confusion, the system also prevents identifiers from using the data type names.

boolean	real
integer	string

Illegal Keywords (used in functions)

Flowgorithm does not permit the names of intrinsic functions to be used.

abs	int	sin	tofixed
arccos	len	size	tointeger
arcsin	log	sqrt	tostring
arctan	log10	tan	toreal
char	random	tochar	
cos	sgn	tocode	

Operators:

Flowgorithm expressions allow the operators used in two major families of programming languages. The "BASIC-family" contains English keywords and operators. The "C-family" (which includes C, Java, C#) is far more symbolic.

Operator	C Family	BASIC Family
Negation	!	not
Modulo	%	mod
Equality	==	=
Inequality	!=	<>
Logical And	&&	and
Logical Or	||	or

Precedence

The following are the precedence levels from high (evaluated first) to low.

Level	Name	Operators	Notes
8	Unary	- ! not	In Visual Basic, "not" precedence level is far lower - above "and", but below all relational operators.
7	Exponent	^	The exponent operator does not exist in C# or Java.
6	Multiply	* / % mod	Division will always be high-precision (floating point)
5	Addition	+ -	In Flowgorithm, "+" will only work with numbers.
4	Concatenate	&	C# and Java use the ambiguous "+" operator for addition and concatenation.
3	Relational	> >= < <= == = != <>	
2	Logical And	and &&	
1	Logical Or	or ||	

Intrinsic Function:

Strings

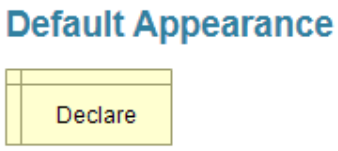
Function	Description
`Len(s)`	Length of a string
`Char(s, i)`	Returns a character from the string ***s*** at index ***i***. Characters are indexed starting at 0.

Data Type Conversion

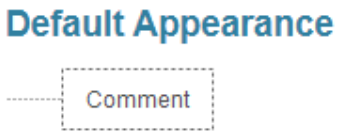
Function	Description
`ToChar(n)`	Convert a character code ***n*** into a character.
`ToCode(c)`	Convert a character ***c*** into a character code (integer).
`ToFixed(r, i)`	Convert real number ***r*** to a string with ***i*** digits after the decimal point. This function is useful for currency.
`ToInteger(n)`	Convert a string to an integer
`ToReal(n)`	Convert a string to an real
`ToString(n)`	Convert a number to a string

Shapes:

1. **Declare Shape:**



2. **Comment Shape**

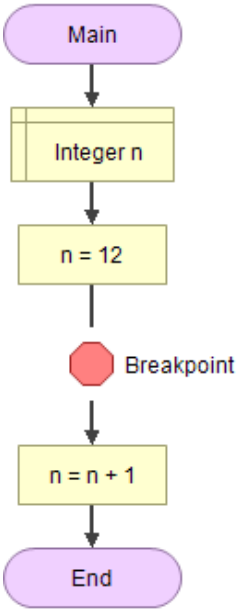


3. **Input Shape**

5. **Output shape**

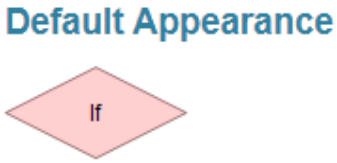
6. **Breakpoint Shape**

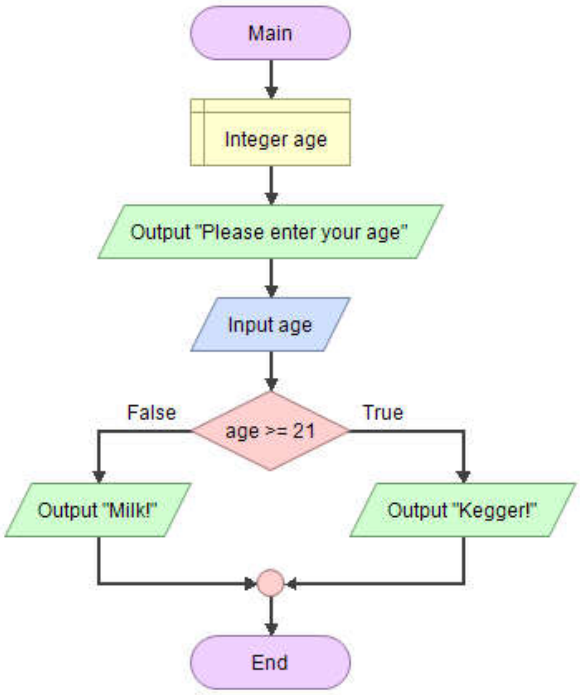
The Breakpoint Shape temporality halt the execution of the program. This is useful both for debugging programs and for demonstrations. Most professional software development applications have some form of the breakpoint.



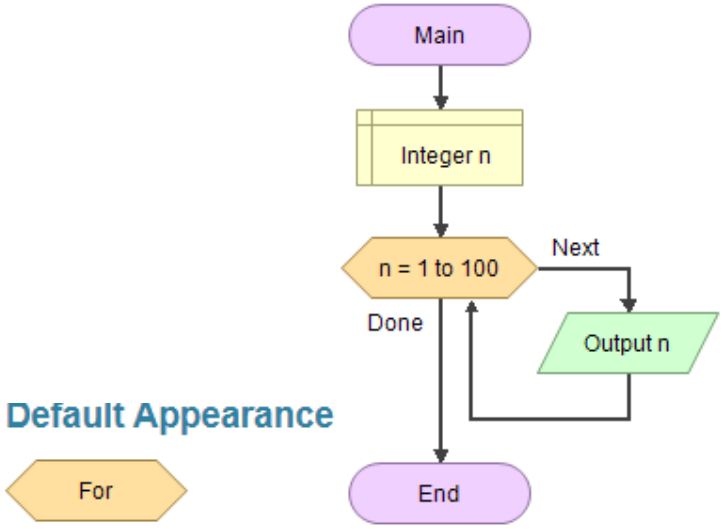
7. **IF Shape:**

An If Statement checks a Boolean expression and then executes a true or false branch based on the result. For example, we want to declare an integer called 'age'. It then reads the age from the keyboard. Finally, an If Statement checks if the age is greater than or equal to 18. Based on this, it either takes the false branch and displays "Sorry, not yet", or takes the true branch and displays "Go vote!".

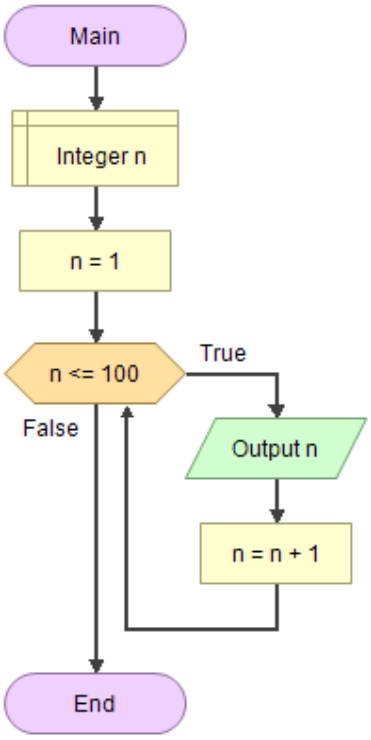




8. **FOR Shape**:



9. **WHILE shape:**



LAB TASK:

1. Apply all the above lab task.
2. Print the table of the no. entered by user.
3. Make a function which prints the table of the number entered by user from starting value to last value. Function must takes 3 inputs x, y, z from the user and pass them to the user.
    - X be the number whose table we want to print
    - Y be the starting value
    - Z be the ending value.

    Make sure ending value must be greater than the starting value