# Algorithm Format: Flowchart

A flowchart models the processing flow of a module, program, or system. It is a graphical representation of the steps that describe the processing performed by a module, program, or system.

## Symbols and Semantics

A flowchart is a graph of connected symbols that describe the processing flow (i.e., control flow) of one module, program or system. The symbols used on a flowchart represent flow, terminal, processing step, input/output, decision, subroutine call, and connectors. An algorithm described using a flowchart starts and ends with a terminal symbol.

|  |  |
| --- | --- |
| **Symbol** | **Semantics** |
|  | A flow is a directed line that connects two symbols within the flowchart.   * The direction of the arrow indicates the processing (i.e., execution) sequence within the flowchart. |
|  | A terminal is an oval labeled with the word “start”, “stop”, “enter”, or “exit”.   * While “start” and “enter” have similar meaning (likewise “stop” and “exit”), convention dictates that “start” and “stop” be used when the flowchart represents the main processing of the module, program or system, and “enter” and “exit” be used when the flowchart represents a sub-processing element (i.e., subroutine) of the module, program, or system. * The “start” and “enter” terminal symbols cannot have an inbound flow and must have only one outbound flow. * The “stop” and “exit” terminal symbols can have one or more inbound flows but no outbound flows. |
|  | A processing step is a rectangle labeled with a verb-noun-phrase that describes the processing performed at this step of the flowchart.   * A processing symbol can have one or more inbound flows but must have only one outbound flow. * After the verb-noun-phrase is performed, processing continues to the symbol pointed to by the outbound flow. |
|  | An input/output is a parallelogram labeled with a verb-noun-phrase that describes an input or output processing statement.   * An input verb-noun-phrase (e.g., “read number”, “get name”) indicates that data is obtained from an external entity. For example, input data may come from a user typing on the keyboard or from a data file. * An output verb-noun-phrase (e.g., “write number”, “display name“) indicates that data is given to an external entity. For example, output data may be displayed to a user or written to a data file. * An input/output symbol can have one or more inbound flows but must have only one outbound flow. * After the verb-noun-phrase is performed, processing continues to the symbol pointed to by the outbound flow. |
|  | A decision is a diamond labeled with a question (aka: test, condition) that must result in a yes/no or true/false answer.   * A decision symbol can have one or more inbound flows but must have exactly two outbound flows. * One outbound flow is labeled with “yes” or “true” while the other outbound flow is labeled with “no” or “false”. * As a result of processing a decision symbol, flow continues to one of the two outbound flows based on the answer to the question. |
|  | A subroutine call is a rectangle with rectangular-sides labeled with a subroutine name that identifies a separate set of processing steps to be performed.   * A subroutine call symbol can have one or more inbound flows but must have only one outbound flow. * After the subroutine is performed, processing continues to the symbol pointed to by the outbound flow. |
|  | A connector is a circle or boxed-arrow labeled with a letter that identifies a separate set of processing steps to be performed.   * A connector symbol can either have one or more inbound flows or one outbound flow. |

## Flowchart Examples (Topics 1-4)

|  |  |
| --- | --- |
| **Flowchart** | **Python Code** |
| # Write a program to calculate the volume and surface area of a sphere from its radius, given as input. | |
| v2-171-06b-Flowchart-ch3-1.wmf | import math  radius = input("Enter radius of sphere: ")  radius = eval(radius)  volume = 4 / 3 \* math.pi \* radius \*\* 3  area = 4 \* math.pi \* radius \*\* 2  print("sphere volume is", volume, \  "and area is", area) |
| # Draw a square using Turtle graphics. | |
| 04-Flowchart-drawSquare.wmf | import turtle  wn = turtle.Screen()  alex = turtle.Turtle()  for i in [0,1,2,3]:  alex.forward(50)  alex.left(90)  wn.exitonclick() |

## Flowchart Examples (Topics 5-6)

|  |  |
| --- | --- |
| **Flowchart** | **Python Code** |
| # Display the lyrics to "Happy Birthday", given a person’s name. | |
| v2-171-06b-Flowchart-ch6-happy-4.wmf | def happy():  print("Happy birthday to you!")  def sing(person):  happy()  happy()  print("Happy birthday, dear", person + ".")  happy()  def main():  sing("Fred")  print()  sing("Lucy")  print()  sing("Elmer") |

## Flowchart Examples (Topics 7-9)

|  |  |
| --- | --- |
| **Flowchart** | **Python Code** |
| # Display a warning message depending on the current temperature (version 1). | |
|  | # 07-tempWarning-1.py  #pre: User ready to enter a temperature.  #post: Returned the entered temperature value.  def getTemperature(prompt):  return eval(input(prompt))  #pre: degrees is a fahrenheit temperature.  #post: A message has been displayed when the  # value of degrees is within a certain  # range.  def tempWarning(degrees):  if degrees > 90:  print(  "It's really hot out there. Be careful!")  if degrees > 65 and degrees < 80:  print("Enjoy the pleasant temperature!")  if degrees < 30:  print("Brrrrr. Be sure to dress warmly!")  def main():  fahrenheit = getTemperature(  "Enter a Fahrenheit temperature: ")  tempWarning(fahrenheit) |
| # Display a warning message depending on the current temperature (version 2). | |
|  | # 07-tempWarning-2.py  #pre: User ready to enter a temperature.  #post: Returned the entered temperature value.  def getTemperature(prompt):  return eval(input(prompt))  #pre: degrees is a fahrenheit temperature.  #post: A message has been displayed when the  # value of degrees is within a certain  # range.  def tempWarning(degrees):  if degrees > 90:  print(  "It's really hot out there. Be careful!")  elif degrees > 65 and degrees < 80:  print("Enjoy the pleasant temperature!")  elif degrees < 30:  print("Brrrrr. Be sure to dress warmly!")  def main():  fahrenheit = getTemperature(  "Enter a Fahrenheit temperature: ")  tempWarning(fahrenheit) |