

P.G DEPARTMENT OF COMPUTER APPLICATIONS

18PCS206- PYTHON PROGRAMMING K4K5 LEVEL QUESTIONS

Unit I

1. Explain the applications of python.
2. Explain the different types of operators with an illustrative program.
3. Explain the scope of variables with a suitable program.
4. Write a program to sort the given words alphabetically and count the vowels.
5. Explain the different types of operators and operator precedence in python with examples

Unit II

6. Elucidate on the built-in functions with examples
7. Elaborate on the math module and the rand and randint functions
8. Create a program to illustrate the function invocation and passing of parameters in functions

Unit III

9. Explain the conditional constructs with a suitable program.
10. Discuss on variable function arguments with an illustrative program.
11. Discuss on Recursion with a suitable program.
12. Explain the for loop with example programs.
13. Explain the String functions with a suitable program.
14. Explain the String methods with a suitable program.

Unit IV

15. Describe the following
 - a. Creating the lists
 - b. Accessing the values in the list
 - c. Updating the list
 - d. Deleting the elements in the list
16. Explain the methods and functions of the Dictionary with a suitable program

17. Examine the operations and methods of the Lists
18. Elaborate the methods and operations of Tuples
19. Compare and contrast the Lists from a tuple using an example program

Unit V

20. Elucidate on modules with suitable program
21. Explain the sys, time n dir functions with a suitable program.
22. Discuss on Exception handling with illustrative examples
23. Explain the menus in the TKinter interface with a suitable program
24. Explain the Frames in TKinter with an illustrative example
25. Describe the process of binding events to widgets with a suitable program

P.G DEPARTMENT OF COMPUTER APPLICATIONS

18PCS206 - PYTHON PROGRAMMING K1 LEVEL QUESTIONS

Unit I

1. All keywords in Python are in
a) lower case b) **UPPER CASE** c) Capitalized d) None of the mentioned
2. Which of the following function returns the max alphabetical character from the string str?
a) lower() b) lstrip() c) **max(str)** d) min(str)
3. Which of the following functions is a built-in function in python?
a) seed() b) sqrt() c) factorial() d) **print()**
4. Is Python case sensitive when dealing with identifiers?
A) Yes b)no c) machine dependent d) none of the mentioned
5. What is the maximum possible length of an identifier?
a) 31 characters b) 63 characters c) 79 characters **d) identifiers can be of any length**
6. Which of the following is not a keyword?
a) eval b) assert c) nonlocal d) pass.
7. What is getattr() used for?
a) **To access the attribute of the object** b) To delete an attribute
c) To check if an attribute exists or not d) To set an attribute
8. What is the output of the following code?
>>>print('new\nline')
a) Error b) Output equivalent to print 'new\nline' c) newline d) **new line**
9. Suppose list1 is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after list1.reverse() ?
a) [3, 4, 5, 20, 5, 25, 1, 3]. b) [1, 3, 3, 4, 5, 5, 20, 25].
c) [25, 20, 5, 5, 4, 3, 3, 1]. d) **[3, 1, 25, 5, 20, 5, 4, 3].**
10. What is the output of the expression:
round(4.576)
a) 4.5
b) 5
c) 4
d) 4.6

Unit -II

11. What is the output of the following function?
any([2>8, 4>2, 1>2])
a) Error
b) True

- c) False
- d) 4>2

12. What is the output of the functions shown below?

```
min(max(False, -3, -4), 2, 7)
```

- a) 2
- b) False**
- c) -3
- d) -4

[View Answer](#)

13. Which is the correct operator for power(x^y)?

- a) X^y
- b) $X**y$**
- c) $X^{^y}$
- d) None of the mentioned

14. Which one of these is floor division in python?

- a) /
- b) //**
- c) %
- d) None of the mentioned

15. What is the output of this expression, $3*1**3$?

- a) 27
- b) 9
- c) 3**
- d) 1

16. Which of these is not a core data type?

- a) Lists
- b) Dictionary
- c) Tuples
- d) Class**

17. Following set of commands are executed in shell, what will be the output?

- a) he**
- b) lo
- c) olleh
- d) hello

18. In python we do not specify types, it is directly interpreted by the compiler, so consider the following operation to be performed.

- a) $x = 13 // 2$
- b) $x = \text{int}(13 / 2)$
- c) $x = 13 \% 2$
- d) All of the mentioned**

19. What error occurs when you execute? `apple = mango`

- a) SyntaxError
- b) NameError**
- c) ValueError
- d) TypeError

20. What is the output of `print 0.1 + 0.2 == 0.3`?

- a) True
- b) False**
- c) Machine dependent
- d) Error

Unit - III

21. What are the outcomes of the functions shown below?

```
sum(2,4,6)
```

```
sum([1,2,3])
```

- a) Error, 6
- b) 12, Error
- c) 12, 6
- d) Error, Error

22. Which of the following is incorrect?

- a) `x = 0b101`
- b) `x = 0x4f5`
- c) `x = 19023`
- d) `x = 03964`

23. What is the output of the following?

```
x=['ab','cd']  
for i in x:  
    i.upper()  
print(x)
```

a) ['ab', 'cd']. b) ['AB', 'CD']. c) [None, None]. d) none of the mentioned

24. What is the output of the following?

```
x='abcd'  
for i in range(len(x)):  
    print(i.upper())
```

a) a b c d b) 0 1 2 3 c) error d) 1 2 3 4

25. What is the output of the following?

```
x='abcd'  
for i in range(len(x)):  
    i.upper()  
    print(x)
```

a) a b c d b) 0 1 2 3 c) error d) none of the mentioned

26. What is the output of the following?

```
string="my name is x"  
for i in string:  
    print(i, end=", ")  
a) m, y, , n, a, m, e, , i, s, , x,  
c) my, name, is, x,
```

b) m, y, , n, a, m, e, , i, s, , x
d) error

27. What is the output of the following?

```
string="my name is x"  
for i in string.split():  
    print(i, end=", ")
```

a) m, y, , n, a, m, e, , i, s, , x, b) m, y, , n, a, m, e, , i, s, , x
c) my, name, is, x, d) error

28. What is the output when following statement is executed ?

```
>>>"a"+"bc"
```

a) a b) bc c) bca d) abc

29. Which of the following statement causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating?

break b) continue c) pass d) None of the above.

30. Which of the following is the use of function in python?

a) **Functions are reusable pieces of programs**
b) Functions don't provide better modularity for your application

- c) you can't also create your own functions
- d) All of the mentioned

Unit-IV

31. Which keyword is use for function?

- a) Fun
- b) Def**
- c) Define
- d) Function

32. What is the type of each element in sys.argv?

- a) set
- b) list
- c) tuple
- d) string**

33. What is the output of the code shown?

```
s=set()  
type(s)
```

- a) <'set'>
- b) <class 'set'>**
- c) set
- d) class set

34. What datatype is the object below ?

L = [1, 23, 'hello', 1].

- a) list**
- b) dictionary
- c) array
- d) tuple

35. What is the output of the following?

```
print('The sum of {0} and {1} is {2}'.format(2, 10, 12))
```

- a) The sum of 2 and 10 is 12**
- b) Error
- c) The sum of 0 and 1 is 2
- d) None of the mentioned

36. Which of the following commands will create a list?

- a) list1 = list()
- b) list1 = [].
- c) list1 = list([1, 2, 3])
- d) all of the mentioned**

37. Suppose list1 is [3, 5, 25, 1, 3], what is min(list1) ?

- a) 3
- b) 5
- c) 25
- d) 1**

38. Suppose list1 is [1, 5, 9], what is sum(list1) ?

- a) 1
- b) 9
- c) 15**
- d) Error

39. To shuffle the list(say list1) what function do we use ?

- a) list1.shuffle()
- b) shuffle(list1)
- c) random.shuffle(list1)**
- d) random.shuffleList(list1)

40. Which of the following is a Python tuple?

- a) [1, 2, 3].
- b) (1, 2, 3)**
- c) {1, 2, 3}
- d) {}

Unit-V

41. What will be the output?

```
1. >>>t=(1, 2, 4, 3, 8, 9)  
2. >>>[t[i]for i in range(0, len(t), 2)]
```

- a) [2, 3, 9].
- b) [1, 2, 4, 3, 8, 9].
- c) [1, 4, 8]**
- d) (1, 4, 8)

42. What will be the output?

```
d={"john":40, "peter":45}
```

a) 40

b) 45

c) "john"

d) "peter"

43. What will be the output?

1. `>>>t =(1, 2)`

2. `>>>2* t`

a) (1, 2, 1, 2)

b) [1, 2, 1, 2]

c) (1, 1, 2, 2)

d) [1, 1, 2, 2]

44. Which of the following statements create a dictionary?

a) `d = {}`

b) `d = {"john":40, "peter":45}`

c) `d = {40:"john", 45:"peter"}`

d) All of the mentioned

45. What will be the output?

1. `d1={"john":40, "peter":45}`

2. `d2={"john":466, "peter":45}`

3. `d1 > d2`

a) True

b) False

c) Error

d) None

46. Suppose `d = {"john":40, "peter":45}`, to delete the entry for "john" what command do we use

a) `d.delete("john":40)`

b) `d.delete("john")`

c) `del d("john":40)` **d) `del d["john"]`.**

46.Dictionary makes use of _____

a) keys, keys

b) keys, key values

c) key values, keys

d) key values, key values

47. Write a list comprehension for number and its cube for:

`l=[1,2,3,4,5,6,7,8,9]`

a) `[x**3 for x in l]`

b) `[x^3 for x in l]`

c) `[x**3 in l]`

d) `[x^3 in l]`

47. Which of the following functions cannot be used on heterogeneous sets?

a) pop

b) remove

c) update

d) sum

48.Which of these definitions correctly describes a module?

a) Denoted by triple quotes for providing the specification of certain program elements

b) Design and implementation of specific functionality to be incorporated into a program

c) Defines the specification of how it is to be used

d) Any program that reuses code

49.Program code making use of a given module is called a _____ of the module.

a) Client

b) Docstring

c) Interface

d) Modularity

50. Which of the statements about modules is false?

a) In the "from-import" form of import, identifiers beginning with two underscores are private and aren't imported

b) `dir()` built-in function monitors the items in the namespace of the main module

c) In the "from-import" form of import, all identifiers regardless of whether they are private or public are imported

d) When a module is loaded, a compiled version of the module with file extension .pyc is automatically produced

P.G DEPARTMENT OF COMPUTER APPLICATIONS

18PCS206-PYTHON PROGRAMMING K3 LEVEL QUESTIONS

Unit I

1. Explain the features of Python programming.
2. Explain how python is compiled and interpreted
3. Write a program to generate the Fibonacci series
4. Explain the different types of operators in python
5. Explain the input, rawinput and print statements
6. Explain the arithmetic operators with a suitable program
7. Explain the special operators with a suitable program
8. Explain the operator precedence and hierarchy in python
9. Explain the statements and expressions with examples
10. Explain the Script mode and interactive modes in python

Unit II

11. Explain the functions from the Math module with an illustrative program
12. How are Random numbers generated? Illustrate it with a program
13. Examine the built-in functions
14. Explain the functions from the math module
15. Discuss on the rand and randint functions
16. Explain the function definition with an illustrative program
17. Explain the different types of arguments a function may assume
18. Explain the void functions and the functions returning values with examples

Unit III

19. Explain the while loop with an illustrative program
20. Explain the nested if statement with an illustrative program
21. Explain the for loops with a suitable program
22. Explain the lambda functions with a suitable program
23. Explain the difference between a pass, break and a continue statements with an example program

24. Explain pattern matching with an illustrative program?
25. Explain the string operators
26. Explain the operations of the string with an illustrative program?
27. Examine the process of comparing the lists using relational operators
28. Write a program to sort the given words alphabetically and count the vowels without using inbuilt functions.
29. Explain the operations of the string with an illustrative program?

Unit IV

30. Explain the operations of a tuple with examples
31. Explain the process of creating iterating and getting values from a dictionary
32. Explain the functions and methods of the tuple with examples
33. Explain the lists with an example program
34. Explain the functions in a tuple with a suitable program
35. Explain the list operations with a suitable program
36. Explain the functions of the tuple with examples
37. Write a python program to concatenate two lists

Unit V

38. Explain the modules and on how to import modules in python with a suitable program
39. Explain the dir() function with examples
40. Explain the Sys function with examples
41. Explain the modules with an illustrative program
42. List out and explain the common Python errors
43. Elaborate on how to handle exceptions with Try/Except/Finally in Python using illustrative examples?
44. Explain Exception handling with a suitable program
45. Examine the features of event driven programming
46. Explain the entry widget with a suitable program
47. Describe the process of creating a CheckButton with a suitable program.
48. Explain the text widget with an example program
49. Explain the Process of creating frames with widgets using a sample program
50. Explain the RadioButton widget with an illustrative program
51. Explain the listboxes with examples
52. Explain the button widget with examples

P.G DEPARTMENT OF COMPUTER APPLICATIONS

18PCS206 PYTHON PROGRAMMING K2 LEVEL QUESTIONS

UNIT I

1.How Python is interpreted?

The python code written as .py is compiled into python bytecode, which creates file with extension .pyc. The .pyc file, created in compilation step, is then executed by appropriate virtual machines. Technically, it's just the last step of what is called the Python interpreter.

2.What happens in the background when you run a Python file?

When we run a .py file, it undergoes two phases. In the first phase it checks the syntax and in the second phase it compiles to bytecode (.pyc file is generated) using Python virtual machine, loads the bytecode into memory and runs.

3.Define keywords?

Keywords are the reserved words in Python. We cannot use a keyword as variable name, function name or any other identifier. They are used to define the syntax and structure of the Python language. In Python, keywords are case sensitive. There are 33 keywords in Python.

4. What are Identifiers?

Identifier is the name given to entities like class, functions, variables etc. in Python. It helps differentiating one entity from another.

5. What is Python Indentation?

Most of the programming languages like C, C++, Java use braces { } to define a block of code. Python uses indentation. A code block (body of a function, loop etc.) starts with indentation and ends with the first unindented line. The amount of indentation is up to you, but it must be consistent throughout that block. Generally four whitespaces are used for indentation and is preferred over tabs.

6. Define Comments in python?

In Python, we use the hash (#) symbol to start writing a comment. It extends up to the newline character. Comments are for programmers for better understanding of a program. Python Interpreter ignores comment.

For multiline comments use triple quotes, either ''' or '''. These triple quotes are generally used for multi-line strings. But they can be used as multi-line comment as well.

7. What are Operators in python?

Operators are special symbols in Python that carry out arithmetic or logical computation. The value that the operator operates on is called the operand. For example:

```
>>> 2+3
```

8. List and explain the Identity operators?

is and is not are the identity operators in Python. They are used to check if two values (or variables) are located on the same part of the memory. Two variables that are equal does not imply that they are identical.

9. What are Membership operators?

In and not in are the membership operators in Python. They are used to test whether a value or variable is found in a sequence (string, list, tuple, set and dictionary). In a dictionary we can only test for presence of key, not the value.

UNIT II

10. What is the purpose of // operator?

"//" is for flooring division

11. How will you convert a string to an int in python?

Python int () method is the standard built-in function to convert a string into an integer value. You call it with a string containing a number as the argument, and it returns the number converted to an actual integer.

12. What are the basic Data Types Supported by Python?

Numeric Data types: int, long, float, NoneType

String: str

Boolean: (True, False)

NoneType: None

13. How many basic types of functions are available in Python?

Python gives us two basic types of functions.

1. Built-in, and
2. User-defined.

14. What is a built-in function that Python uses to iterate over a number sequence?

Range () generates a list of numbers, which is used to iterate over for loops.

15. Define Functions

In Python, function is a group of related statements that perform a specific task. Functions help break our program into smaller and modular chunks. As our program grows larger and

larger, functions make it more organized and manageable. Furthermore, it avoids repetition and makes code reusable.

Syntax of Function

```
def function name(parameters):  
    """docstring"""  
    Statement(s)
```

16. What is “Call by Value” in Python?

In call-by-value, the argument whether an expression or a value gets bound to the respective variable in the function.

Python will treat that variable as local in the function-level scope. Any changes made to that variable will remain local and will not reflect outside the function.

17. What is “Call by Reference” in Python?

We use both “call-by-reference” and “pass-by-reference” interchangeably. When we pass an argument by reference, then it is available as an implicit reference to the function, rather than a simple copy. In such a case, any modification to the argument will also be visible to the caller.

18. Is it mandatory for a Python function to return a value?

It is not at all necessary for a function to return any value. However, if needed, we can use none as a return value.

19. What is the difference between pass and continue in Python?

The continue statement makes the loop to resume from the next iteration.

On the contrary, the pass statement instructs to do nothing, and the remainder of the code executes as usual.

UNIT III

20. Define control flow statement.

A program's control flow is the order in which the program's code executes. The control flow of a Python program is regulated by conditional statements, loops, and function calls

21. Define break statement

The break statement terminates the loop containing it. Control of the program flows to the statement immediately after the body of the loop. If it is inside a nested loop (loop inside another loop), break will terminate the innermost loop.

22. Define recursion

Recursion is the process of defining a problem (or the solution to a problem) in terms of a simpler version of itself. A recursive procedure or routine is one that has the ability to call itself.

23. What is pass in Python?

It is used when a statement is required syntactically but you do not want any command or code to execute.

The **pass** statement is a *null* operation; nothing happens when it executes. The **pass** is also useful in places where your code will eventually go, but has not been written yet (e.g., in stubs for example)

24. What is a Lambda Function?

In Python, anonymous function is a function that is defined without a name. While normal functions are defined using the `def` keyword, in Python anonymous functions are defined using the `lambda` keyword. Hence, anonymous functions are also called lambda functions.

Lambda Functions

A lambda function has the following syntax.

Syntax of Lambda Function

Lambda arguments: expression

25. What are the principal differences between the lambda and def?

`Def` can hold multiple expressions while `lambda` is a uni-expression function. `Def` generates a function and designates a name to call it later. `Lambda` forms a function object and returns it. `Def` can have a return statement. `Lambda` can't have return statements. `Lambda` supports to get used inside a list and dictionary.

26. What is a continue statement?

The `continue` statement is used to skip the rest of the code inside a loop for the current iteration only. Loop does not terminate but continues on with the next iteration.

27. What are events and callbacks?

In **Python**, a **callback** is simply a function or a method passed to `LocalSolver`. A **callback** takes two parameters: the `LocalSolver` object that triggers the **event** and the type of the **callback**. It is possible to use the same **callback** method or object for multiple events or multiple `LocalSolver` instances

28. What are Strings?

String is sequence of Unicode characters. We can use single quotes or double quotes to represent strings. Multi-line strings can be denoted using triple quotes, `'''` or `"""`.

29. How will you replaces all occurrences of old substring in string with new string?

Return Value from `replace()`. The `replace()` method returns a copy of the string where old substring is replaced with the new substring. The original string is unchanged. If the old substring is not found, it returns the copy of the original string.

30. What is slicing in Python?

Slicing is a string operation for extracting a part of the string, or some part of a list. In Python, a string (say text) begins at index 0, and the nth character stores at position text[n-1]. Python can also perform reverse indexing, i.e., in the backward direction, with the help of negative numbers. In Python, the slice() is also a constructor function which generates a slice object. The result is a set of indices mentioned by range (start, stop and step). The slice() method allows three parameters. 1. start – starting number for the slicing to begin. 2. stop – the number which indicates the end of slicing. 3. step – the value to increment after each index (default = 1).

31. What does the len() function do in Python?

In Python, the len() is a primary string function. It determines the length of an input string.

```
>>>some_string = 'techbeamers'  
>>>len(some_string)
```

32. What is Rstrip() in Python?

Python provides the rstrip() method which duplicates the string but leaves out the whitespace characters from the end.

The rstrip() escapes the characters from the right end based on the argument value, i.e., a string mentioning the group of characters to get excluded.

33. How do you use the split() function in Python?

Python's split() function works on strings to cut a large piece into smaller chunks, or sub-strings. We can specify a separator to start splitting, or it uses the space as one by default.

UNIT IV

34. Define Lists

List is an ordered sequence of items. It is one of the most used datatype in Python and is very flexible. All the items in a list do not need to be of the same type. Declaring a list is pretty straightforward. Items separated by commas are enclosed within brackets [].

35. Define Tuples

Tuple is an ordered sequence of items same as list. The only difference is that tuples are immutable. Tuples once created cannot be modified. They are used to write-protect data and are usually faster than list as it cannot change dynamically. Tuple is defined within parentheses () where items are separated by commas.

36. When do you choose a list over a tuple?

When there is an immutable ordered list of elements we choose tuple. Because we cannot add/remove an element from the tuple. On the other hand, we can add elements to a list using append() or extend() or insert(), etc., and delete elements from a list using remove() or pop().

Simple tuples are immutable, and lists are not. Based on these properties one can decide what to choose in their programming context.

37. How do you get the last value in a list or a tuple?

When we pass -1 to the index operator of the list or tuple, it returns the last value. If -2 is passed, it returns the last but one value.

38. What is the difference between Python append () and extend () functions?

The extend() function takes an iterable (list or tuple or set) and adds each element of the iterable to the list. Whereas append takes a value and adds to the list as a single object.

39. Define a Set

Set is an unordered collection of unique items. Set is defined by values separated by comma inside braces { }. Items in a set are not ordered.

40. What is a Dictionary?

Dictionary is an unordered collection of key-value pairs. It is generally used when we have a huge amount of data. Dictionaries are optimized for retrieving data. We must know the key to retrieve the value. In Python, dictionaries are defined within braces { } with each item being a pair in the form key: value. Key and value can be of any type.

41. How do you create a dictionary in Python?

Let's take the example of building site statistics. For this, we first need to break up the key-value pairs using a colon(":"). The keys should be of an immutable type, i.e., so we'll use the data-types which don't allow changes at runtime. We'll choose from an int, string, or tuple. However, we can take values of any kind. For distinguishing the data pairs, we can use a comma(",") and keep the whole stuff inside curly braces({...}).

42. How do you read from a dictionary in Python?

To fetch data from a dictionary, we can directly access using the keys. We can enclose a "key" using bracket [...] after mentioning the variable name corresponding to the dictionary.

```
>>>site_stats={'site':'tecbeamers.com','traffic':10000,"type":"organic"}
>>>print(site_stats["traffic"])
```

43. Differentiate a list from a tuple?

The main **difference** between **lists** and a **tuples** is the fact that **lists** are mutable whereas **tuples** are immutable. A mutable data type means that a python object of this type can be modified.

44.How do you check the presence of a key in a dictionary?

We can use Python's "in" operator to test the presence of a key inside a dict object.

UNIT V

45. Define Python Modules

Modules refer to a file containing Python statements and definitions. A file containing Python code, for e.g.: example.py, is called a module and its module name would be example. We use modules to break down large programs into small manageable and organized files. Furthermore, modules provide reusability of code. We can define our most used functions in a module and import it, instead of copying their definitions into different programs.

46. What is the use of import keyword?

We can import the definitions inside a module to another module or the interactive interpreter in Python. We use the import keyword to do this.

47. What is the use of dir() built-in function?

We can use the dir() function to find out names that are defined inside a module.

48. Define exceptions

Errors can also occur at runtime and these are called exceptions. They occur, for example, when a file we try to open does not exist (FileNotFoundError), dividing a number by zero (ZeroDivisionError), module we try to import is not found (ImportError) etc. Whenever this type of runtime error occurs, Python creates an exception object.

Python has many built-in **exceptions** which forces your program to output an **error** when something in it goes wrong. ... In **Python**, users can **define** such **exceptions** by creating a new class. This **exception** class has to be derived, either directly or indirectly, from **Exception** class.

49. How to catch Exceptions in Python?

In Python, exceptions can be handled using a try statement. A critical operation which can raise exception is placed inside the try clause and the code that handles exception is written in except clause.

50. What is the use of raise keyword?

In Python programming, exceptions are raised when corresponding errors occur at run time, but we can forcefully raise it using the keyword raise. We can also optionally pass in value to the exception to clarify why that exception was raised.

51. What is the use of finally keyword?

The try statement in Python can have an optional finally clause. This clause is executed no matter what, and is generally used to release external resources. For example, we may be connected to a remote data center through the network or working with a file or working with a Graphical User Interface (GUI). In all these circumstances, we must clean up the resource once used, whether it was successful or not. These actions (closing a file, GUI or disconnecting from network) are performed in the finally clause to guarantee execution.

52. What is a root component?

Root is the parent window. All applications have a “root” window. This is the parent of all other widgets. Only one root window should be created

53. What is a Grid?

Grid is a layout or geometry manager. It is responsible for determining where widgets go in a window, and what happens as the window changes size, widgets get added, removed, etc...

54. Define widgets

Widgets are standard graphical user interface (GUI) elements, like different kinds of buttons and menus.

What is Event driven programming

– a programming paradigm where the flow of the program is driven by sensor outputs or user actions(Events)