**– Question Bank and Solution --**

**Q1. Explain any six features of Python programming. [4]**

**Ans:**

1. Easy to code: Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc.

2. Free and Open Source: Python language is freely available at the official website and you can download it from the given download link below click on the Download Python keyword.

3. Object-Oriented Language: One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects encapsulation, etc.

4. GUI Programming Support: Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python..

5. High-Level Language: Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6. Extensible feature: Python is a Extensible language. We can write us some Python code into C or C++ language and also we can compile that code in C/C++ language.

**Q2.What is a problem? Explain six problem solving steps.** **[4]**

**Ans:**

A problem is a state of difficulty that need to be resolved, when a problem exist then means there is some uncertainty in the solution. While solving the problem there is a desire to attain some specific goal.

The Six Steps

1. Define the Problem or identify the problem
2. Determine/understand the Root Cause(s) of the Problem
3. Identify the Alternative Solutions
4. Select the best way to Solution
5. List the instruction using the selected Solution
6. Evaluate the Outcome

**Q3. Explain following terms with suitable examples. [6]**

**Ans:**

1. **Comment**

Comments are non-executable statements in program. They are just added to describe the statements in the program code. All characters following by # consider as comments. A program can have any number of comments.

Ex.

#This is a comment  
print("Hello, World!")Reserve Words

**ii) Reserved words:** Reserved words (also called keywords) are defined with predefined meaning and syntax in the language. These keywords have to be used to develop programming instructions. Reserved words can’t be used as identifiers for other programming elements like name of variable, function etc.

1. **Identifier:**

Identifiers Known as name given to identity something. This something can be variable, function, class, module or other user defined objects.

For naming there are some basic rules,

First character must be underscore \_ or letter( alphabets ).

Rest of the identifier name can be \_ , letter or numbers.

Identifier names are case-sensitive

Any other symbols are not allowed with in identifiers

**Q4.Write a program to largest in three numbers. [5]**

**Ans:**

a=9

b=74

c=87

if (a>b):

if (a>c):

print("Print a is largest...")

else :

print("Print c is largest...")

elif (b>c):

print("Print b is largest...")

else:

print("Print c is largest...")

**Q5.What is identifier? List the rule to rules to name an identifier. [4]**

**Ans:**

Python Identifier is the name we give to identify a variable, function, class, module or other object. That means whenever we want to give an entity a name, that’s called identifier.

Sometimes variable and identifier are often misunderstood as same but they are not.

For naming there are some basic rules,

* First character must be underscore \_ or letter( alphabets ).
* Rest of the identifier name can be \_ , letter or numbers.
* Identifier names are case-sensitive
* Any other symbols are not allowed with in identifiers

**Q6.List down types of operators in Python. Explain relational operators.**  **[5]**

**Ans:**

Python language supports the following types of operators.

1. Arithmetic Operators
2. Comparison (Relational) Operators
3. Assignment Operators
4. Logical Operators
5. Bitwise Operators
6. Membership Operators
7. Identity Operators

Relational operators are used for comparing the values. It either returns True or False according to the condition. These operators are also known as Comparison Operators.

| **Operator** | **Description** | **Syntax** |
| --- | --- | --- |
| > | Greater than: True if the left operand is greater than the right | x > y |
| < | Less than: True if the left operand is less than the right | x < y |
| == | Equal to: True if both operands are equal | x == y |
| != | Not equal to – True if operands are not equal | x != y |
| >= | Greater than or equal to: True if left operand is greater than or equal to the right | x >= y |
| <= | Less than or equal to: True if left operand is less than or equal to the right | x <= y |

**Q7.Explain different data type supported by python with example. [6]**

**Ans:**

In programming, data type is an important concept. Variables can store data of different types, and different types can do different things.

Python has the following data types built-in by default, in these categories:

**Numeric data types:**

Numeric data type represents the data which has numeric value.

Integers – contains positive or negative whole numbers without fraction or decimal.

Ex. A=5

<class 'int'>

Float –It is a real number with floating point representation. It is specified by a decimal point.

Ex. A=5.0.

<class 'float'>

Complex Numbers – Complex number is represented by complex class. It is specified as (real part) + (imaginary part)j.

Ex. A= 2+3j

<class 'complex'>

**Sequence types:**

In Python, sequence is the ordered collection of similar or different data types.

String

A string is a collection of one or more characters. String is created by put in a single quote, double-quote or triple quote.

Ex. str1=“snjb coe”

<class 'str'>

List

Lists are just like the arrays; list is an ordered collection of data. It is very flexible as the items in a list do not need to be of the same type.

Ex.L1= [“Mango”, 100, True, 3.65]

<class 'list'>

Tuple

Just like list, the only difference between tuple and list is that tuples are immutable i.e. tuples cannot be modified after it is created. It is represented by tuple class.

Ex.T1= (“Apple”, 543, False, 87.63)

<class 'tuple'>

**Boolean type:**

Data type with one of the two built-in values, True or False. Boolean objects that are equal to True are truthy (true), and those equal to False are falsy (false).

Ex. A=True

<class 'bool'>

**Set type:**

In Python, Set is an unordered collection of data type that is iterable, mutable and has no duplicate elements.

Ex. set1 = set([1, 2, 'Geeks', 4, 'For', 6, 'Geeks'])

<class 'set'>

**Dictionary type:**

Dictionary used to store data values like a map, Dictionary holds key: value pair. Key-value is provided in the dictionary to make it more optimized.

Ex. Dict = {1: 'Geeks', 2: 'For', 3: 'Geeks'}

<class 'dict'>

**Q8.Write an algorithm to find sum of 'n' natural number [5]**

**Ans:**

Step 1: Start

Step 2: input number is n, Initialize counter and sum by 0

Step 3: Loop until counter<=number

Add counter in sum

Increment counter by 1

Step 4: Display sum

Step 5: Stop

**Q9.Explain flow-chart and algorithm with example.** **[6]**

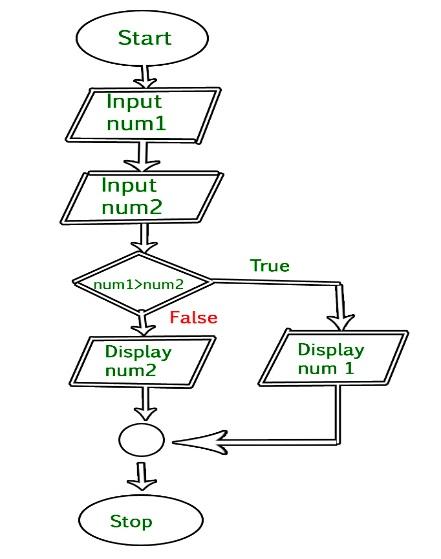
**Ans:**

Flowchart is a graphical representation of an algorithm. Programmers often use it as a program-planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information and processing.

The process of drawing a flowchart for an algorithm is known as “flowcharting”.

Example:

**A** **flowchart to input two numbers from user and display the larger of two numbers**



**Q10. Write a program to generate a Fibonacci series of ‘n’ numbers.** **[6]**

**Ans:**

#Python program to generate Fibonacci series until 'n' value

num = int(input("Enter the value of 'n': "))

n1, n2 = 0, 1

print("Fibonacci Series:", n1, n2, end=" ")

for i in range(2, num):

n3 = n1 + n2

n1 = n2

n2 = n3

print(n3, end=" ")

print()

**Q11.Describe the following term with examples i) break ii) continue iii) pass [6]**

**Ans:**

* 1. **Break**

**Ans:**

The break statement in Python terminates the current loop and resumes execution at the next statement, just like the traditional break found in C.

The most common use for break is when some external condition is triggered requiring a hasty exit from a loop. The break statement can be used in both while and for loops.

Example:

for letter in 'Python': # First Example

if letter == 'h':

break

print 'Current Letter :', letter

* 1. **Continue**

**Ans:**

The continue statement in Python returns the control to the beginning of the while loop. The continue statement rejects all the remaining statements in the current iteration of the loop and moves the control back to the top of the loop. The continue statement can be used in both while and for loops.

Example:

var = 10 # Second Example

while var > 0:

var = var -1

if var == 5:

continue

print 'Current variable value :', var

print "Good bye!"

* 1. **Pass**

**Ans:**

The pass statement in Python 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.

Example:

for letter in **'Python'**:

if letter == 'h':

**pass**

print 'This is pass block'

print 'Current Letter :', letter

print "Good bye!"

**Q12.Explain selection/conditional statements in python. [4]**

**Ans:**

* 1. **if statement**

The if statement contains a logical expression using which data is compared and a decision is made based on the result of the comparison. If the boolean expression evaluates to TRUE, then the block of statement(s) inside the if statement is executed. If boolean expression evaluates to FALSE, then the first set of code after the end of the if statement(s) is executed.

Syntax:

if expression:

statement(s)

Example :

var1 = 100

if var1:

print "1 - Got a true expression value"

print var1

* 1. **if. else statement**

An else statement can be combined with an if statement. An else statement contains the block of code that executes if the conditional expression in the if statement resolves to 0 or a FALSE value.

Syntax:

if expression:

statement(s)

else:

statement(s)

Example:

var1 = 100

if var1:

print "1 - Got a true expression value"

print var1

else:

print "1 - Got a false expression value"

print var1

* 1. **if..elif..else statement**

The elif statement allows you to check multiple expressions for TRUE and execute a block of code as soon as one of the conditions evaluates to TRUE.

Syntax:

if expression1:

statement(s)

elif expression2:

statement(s)

else:

statement(s)

Example:

var = 100

if var == 200:

print "1 - Got a true expression value"

print var

elif var == 150:

print "2 - Got a true expression value"

print var

else:

print "3 - Got a false expression value"

print var

print "Good bye!"

**Q13.Write a program to print even numbers between range 1 to 15 using "while loop". [5]**

**Ans:**

num=1

while (num<=15):

if (num%2==0):

print(num)

num=num+1

**Q14.What dictionary? How to add and remove elements in dictionary? [6]**

**Ans:**

A dictionary is a kind of data structure that stores items in key-value pairs. A key is a unique identifier for an item, and a value is the data associated with that key.

Dictionaries often store information such as words and definitions, but they can be used for much more.

Dictionaries are mutable in Python, which means they can be changed after they are created. The items in a dictionary are not stored in any particular order.

**Syntax**

dictionary\_name = {key\_1: value\_1, key\_2: value\_2, key\_3: value\_3…}

**Add elements in dictionary**

Adding an item to the dictionary is done by using a new index key and assigning a value to it

thisdict = {"brand":"Ford", "model":"Mustang", "year":1964}  
thisdict["color"] = "red"  
print(thisdict)

**Remove elements from dictionary**

We can use the following methods to remove items from a dictionary in Python:

1. **The del keyword**

The del keyword method uses the keyword from the dictionary to remove an item

thisdict = {"brand":"Ford", "model":"Mustang", "year":1964}

del thisdict["model"]

print(thisdict)

1. **The clear() method**

The clear() method clears all the items in the dictionary.

countries={"Ghana": "Accra", "China": "Beijing"}

countries.clear()

print(countries)

1. **The pop() method**

The pop method uses the keyword from the given dictionary to remove an item from that dictionary. The pop method is similar to the del keyword method because they both use the keyword from the dictionary.

thisdict = {"brand":"Ford", "model":"Mustang", "year":1964}

thisdict.pop("model")

print(thisdict)

1. **The popitem() method**

The popitem() removes the last item added to the dictionary.

countries={"Ghana": "Accra", "China": "Beijing"}

countries.popitem()

print(countries)

**Q15. Explain while loop with flowchart. [4]**

**Ans:**

The while statement is used to execute a set of statements repeatedly. The while statement is also known as entry control loop statement because the condition is tested before any of the statements in the statement block is executed. If the condition is true, only then statements will executed otherwise if the condition is false, control will jump to next statement (out of loop).

Syntax:

while expression:

statement(s)

Example:

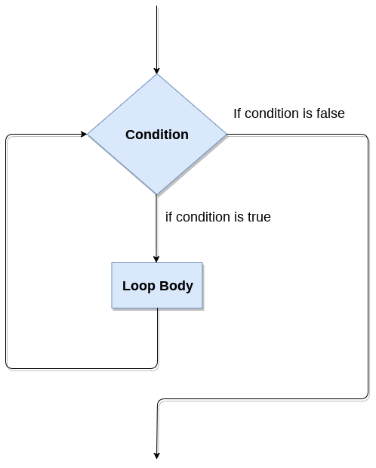
count = 0

while (count < 9):

print 'The count is:', count

count = count + 1

print "Good bye!"

Flowchart: 

**Q15.Write a program in python to find factorial of a number using for loop. [5]**

**Ans:**

num=int(input("Enter the number : "))

fact=1

if (num==0):

pass

else:

for i in range (1,num+1):

fact=fact\*i

print("Factorial or number: ",fact)

**Q16.What is a list? Explain any three operations of list.** **[5]**

**Ans:**

The list is a most versatile datatype available in Python which can be written as a list of comma-separated values (items) between square brackets.

Operation in List

1. cmp() Operation:

Python list method cmp() compares elements of two lists. If elements are of the same type, perform the compare and return the result. If elements are different types, check to see if they are numbers

Syntax:

cmp(list1, list2)

Example:

list1, list2 = [123, 'xyz'], [456, 'abc']

print cmp(list1, list2)

print cmp(list2, list1)

list3 = list2 + [786];

print cmp(list2, list3)

1. len() Operation:

Python list method len() returns the number of elements in the list.

Syntax:

len (list)

Example:

list1, list2 = [123, 'xyz', 'zara'], [456, 'abc']

print "First list length : ", len(list1)

print "Second list length : ", len(list2)

1. max() Operation: Python list method max returns the elements from the list with maximum value.

Syntax:

max (list)

Example:

list1, list2 = [123, 'xyz', 'zara', 'abc'], [456, 700, 200]

print "Max value element : ", max(list1)

print "Max value element : ", max(list2)

**Q17.Write a program to test whether a number entered by the user is positive, negative or zero.** **[5]**

**Ans :**

n = float(input("Input a number: "))

if n >= 0:

if n == 0:

print("It is Zero!")

else:

print("Number is Positive number.")

else:

print("Number is Negative number.")