Dezyne École College

Bachelor of Computer Applications (BCA)

Third Year-5th Semester Python Programming

Probable Question for Python Programming

Unit - 1

Q1. What are keywords in Python? Give any 5 examples and explain why they cannot be used as identifiers.

Q2. Define identifiers in Python. What are the rules for naming identifiers? Give valid and invalid examples.

Q3. What is indentation in Python? Why is it important? What happens if indentation is incorrect?

Q4. Explain dynamic typing in Python with an example. How is it different from static typing?

Q5. Differentiate between mutable and immutable data types. Give 3 examples of each.

Q6. What are built-in type conversion methods in Python? Explain int(), float(), str(), list() with examples.

Q7. List all arithmetic operators in Python. Explain the difference between / and // operators.

Q8. What are identity operators? Explain the difference between 'is' and '==' operators with examples.

Q9. Explain all comparison operators in Python with examples. How do they work with different data types?

Q10. What are logical operators in Python? Explain their precedence and truth table with examples.

Q11. Describe membership operators in Python. How do they work with strings, lists, and tuples? Give examples.

Q12. Explain the difference between break, continue, and pass statements. In which scenarios would you use each?

Q13. What is the range() function in Python? Explain its syntax with different parameters. Give examples of range(5), range(2,8), and range(1,10,2).

Q14. Explain nested if-else statements. What are the advantages and disadvantages? Write the general syntax.

Q15. What are nested loops in Python? Explain with syntax. What is the time complexity consideration for nested loops?

Q16. Explain different ways to create arrays/lists in Python. What are the advantages of using lists over other data structures?

Q17. Compare and contrast all control statements in Python: Conditional statements (if, ifelse, elif, nested if-else). Explain the syntax, flow of execution, and use cases for each. Discuss when to use which type of conditional statement. Provide the general algorithm for decision making in programming.

Q18. Discuss looping structures in Python in detail: While loop vs For loop - syntax, use cases, advantages. When to use while loop over for loop and vice versa. Explain loop control statements (break, continue, pass). Discuss infinite loops and how to avoid them. Explain the concept of loop else clause.

Q19. Explain operators in Python comprehensively: Classify all types of operators with examples. Discuss operator precedence and associativity. Explain how operators work with different data types. What happens when operators are used with incompatible data types? Discuss short-circuit evaluation in logical operators.

Q20. Discuss arrays/lists in Python thoroughly: What are arrays? How are they implemented in Python as lists? Explain all basic operations: creation, traversal, insertion, deletion, search, update. Discuss the time complexity of each operation. Compare list operations with other data structures. Explain memory management and dynamic resizing of lists. What are the advantages and limitations of using lists?

Q21. Predict the output:

```
print(True + True)
print(True * False)
print(False - True)
print(True / True)
```

Q22. Predict the output:

```
a = [1, 2, 3, 4, 5]
print(a[1:4])
print(a[-2:])
print(a[::2])
print(a[::-1])
```

Q23. Predict the output:

```
def mystery():
    a = [1, [2, 3],4]
    b = a.copy()
    c = a[:]
    d = list(a)
    a[1].append(5)
    a[0] = 10
    print("a:", a)
    print("b:", b)
    print("c:", c)
    print("d:", d)

mystery()
```

Q24. Write a program that creates the following pattern for n = 4:

1 0 0 0
0 2 0 0
0 0 3 0
0 0 0 4
4 0 0 0
0 3 0 0
0 0 2 0
0 0 0 1

Q25. Write a program that takes a string and creates a new string where:

- Every vowel is replaced by the next vowel (a→e, e→i, i→o, o→u, u→a)
- Every consonant is replaced by its position in alphabet
- Spaces remain unchanged

Q26. Create a program that merges two sorted arrays into one sorted array without using built-in sort functions. Handle arrays of different sizes.

Q27. Write a program that finds all pairs of numbers in an array whose sum equals a target value. Don't use the same element twice and avoid duplicate pairs.

Unit -2

Q1. What is the difference between mutable and immutable data types in Python?

Q2. How does string indexing work with negative indices?

Q3. What is the purpose of escape sequences in strings?

Q4. Explain the difference between find() and index() methods for strings.

Q5. What happens when you use upper() on a string that contains numbers?

Q6. How does isalpha() behave with Unicode characters?

Q7. What is the difference between append() and extend() methods in lists?

Q8. Explain how pop() method works without arguments versus with index argument.

Q9. What is the time complexity of insert() operation in a list?

Q10. How do you implement a stack using Python lists?

Q11. What is the difference between remove() and discard() methods in sets?

Q12. Can you store mutable objects like lists in a set? Why or why not?

Q13. What is set comprehension and how is it different from list comprehension?

Q14. Explain the difference between union (|) and update operations in sets.

Q15. How do dictionary keys need to be structured and why?

Q16. What happens when you access a non-existent key in a dictionary?

Q17. Explain the difference between pop() and popitem() methods in dictionaries.

Q18. How does dictionary ordering work in Python 3.7+?

Q19. What is the purpose of setdefault() method in dictionaries?

Q20. How do you create a shallow copy versus deep copy of a dictionary?

Q21. What is the difference between del statement and clear() method?

Q22. Explain how slicing works with step parameter in lists and strings.

Q23. What is the difference between sort() and sorted() functions?

Q24. How do you reverse a list in-place versus creating a new reversed list?

Q25. What is the difference between is and == operators when comparing strings?

Q26. Predict Output

```
s = "aBc123XyZ"
lst = []
for i, char in
enumerate(s):
    if char.isdigit():
        lst.append(int(char))
    elif char.isupper():
        lst.append(char.lower())
    else:
        lst.append(char.upper())
result = ''.join(str(x) for x in
lst)
print(result)
print(s.swapcase())
print(result == s.swapcase())
```

Q27. Predict Output

```
d = {'x': {1, 2, 3}, 'y': {2, 3,
4}}
s = set()
for key in d:
    s = s.union(d[key])
    d[key] = d[key].intersection({2, 3, 5})
d['z'] =
s.difference(d['x'].union(d['y']))
print(len(s))
print(d['x'])
print(sorted(d['z']))
```

Q28. Predict Output

```
original = [1, [2, 3], 4, [5,6]]
copy1 = original[:]
copy2 = original.copy()
original[1].append(7)
original[0] = 10
copy1[3] = [8, 9]
print(original[1])
print(copy1[1])
print(copy2[1])
print(copy1[3])
```

Q29. Multi-Level Dictionary Analyzer Write a program that takes a nested dictionary representing student data and performs these operations:

1. Calculate the average grade for each student across all subjects
2. Find students who have the same set of subjects
3. Create a new dictionary grouped by grade ranges: 'A' (90-100), 'B' (80-89), 'C' (70-79), 'D' (60-69), 'F' (<60)
4. Return statistics including total students, subject-wise averages, and grade distribution

Input format: {'student1': {'math': 85, 'science': 92}, 'student2': {'math': 78, 'english': 88}}

Unit – 3

Q1. What is the difference between a required argument and a keyword argument in Python functions?

Q2. Explain the concept of default arguments with an example.

Q3. What is a variable-length argument? Explain *args and **kwargs.

Q4. Define an anonymous function. How is it different from a named function?

Q5. Differentiate between local and global variables with examples.

Q6. What is the use of the global keyword in Python?

Q7. Explain recursion with an example. What is the base case in recursion?

Q8. Write the syntax of a lambda function. When would you prefer to use it?

Q9. What is the purpose of the dir() function in Python?

Q10. How are modules imported in Python? What are the different ways of importing?

Q11. Differentiate between a module and a package.

Q12. How can we install an external package in Python

Q13. List three built-in modules and mention one function each from them.

Q14. Explain the use of the math, random, and statistics modules.

Q15. What is an exception? How does it differ from an error?

Q16. Explain the structure and purpose of try, except, else, and finally blocks.

Q17. How can you manually raise an exception? Give an example.

Q18. What is an invoked function in the context of exceptions?

Q19. Write the steps to open a file and read its contents in Python.

Q20. What is the difference between 'r', 'w', 'a', and 'rb' file modes?

Q21. How do you rename and delete a file in Python?

Q22. What are file positions? How can they be changed using seek() and tell()?

Q23. Explain how dictionaries can be used with file operations.

Q24. What are tuples? How are they different from lists?

Q25. How can a tuple return multiple values from a function?

Q26. Predict the output

```
def func(x, y=[]):
    y.append(x)
    return y
print(func(1))
print(func(2))
print(func(3))
```

Q27. Predict the output

```
def add(a, b):
   return a + b
x = lambda a, b: a * b + add(a, b)
print(x(2, 3))
```

Q28. Predict the output

```
def outer():
   x = "outer"
   def inner():
      nonlocal x
      x = "inner"
      print("Inner:", x)
   inner()
   print("Outer:", x)
outer()
```

Q29. Write a program to sort a list of tuples based on the second value using a lambda function.
Input: [('a', 3), ('b', 1), ('c', 2)]
Output: [('b', 1), ('c', 2), ('a', 3)]

Q30. Write a function that takes a list of numbers and returns a tuple with the following: Minimum value, Maximum value, Average value Return all three using a tuple.

Q31. Generate a list of 100 random integers between 1 and 50 using the random module. Then use the statistics module to calculate and print: Mean, Median and Mode.

Q32. Write a program that reads a text file and counts the frequency of each word, storing it in a dictionary and printing the result sorted by frequency.

Total No. of Pages: 02

[Total No. of Questions: 18]

1691

BCA (Part-II) Examination, 2023
Paper - BCA - 202
Python Programming

Duration of Examination: 3 Hours
परीक्षा की अवधि: 3 घण्टा

Max. Marks: 50
पूर्णांक: 50

Instructions to the Candidates:
परीक्षार्थी के लिए निर्देश:-

Part-A (Compulsory)

Answer all ten questions (upto 20 words each). Each question carries equal marks. *(Marks-15)*

Part-B (Compulsory)

Answer all five questions (upto 50 words each). Each question carries equal marks. *(Marks-15)*

Part-C

Answer all three questions (upto 400 words each). Three question of 7, 7 & 6 marks. *(Marks-20)*

Part-A (Compulsory)

1- Who developed Python language?
2- What is the difference between append and extend ( ) .
3- What is dynamic typing?
4- Define Keyword.
5- Is Python case sensitive?
6- What is negative indexing in list?
7- What is string slicing?
8- What is docstring?
9- Define Dictionary.
10- What is the difference between *a=(10)* and *a=(10,)* in Python?

Part-B (Compulsory)

11- What is the difference between local variable and global variable?
12- What is the difference between pickling and unpickling?
13- What is the difference between mutable and immutable data types?
14- What is range ( ) function.
15- Write a program in Python to check whether the entered number is prime or not.

1691

Part-C

Unit-I

16- Explain the features of Python? What are its application areas? Write its advantages also.

OR

Explain different types of operators with example.

Unit-II

17- Explain any seven built- in- methods of python string.

OR

What is function? What are its types? What are the types of arguments in it.

Unit-III

18- What are python modules. How to import it. Explain some built in modules that are often used in it.

OR

What are the different file processing modes supported by python? Explain.

* * * * *