

Question Bank for the subject Python Application Programming (15CS664)

Module 1:

1. Describe the hardware architecture of a computer.
2. What is a program? Briefly discuss building blocks of a program.
3. Bring out the difference between an Interpreter and a Compiler.
4. What are various ways in which a Python program can be executed? Discuss.
5. What are the rules for naming variables in Python?
6. Write a short note on:
 - i) Comments in Python
 - ii) `**` and `//` operators
 - iii) Short-circuit evaluation
 - iv) `input()` Function
 - iv) Boolean values
 - v) Python keywords
 - vi) Types of Errors
 - vii) Precedence of Operators
 - viii) Built-in Functions
 - ix) Random Numbers
 - x) Math functions
7. When is a nested condition used? Discuss with an example.
8. Discuss exception handling in Python with an example.
9. What are fruitful functions and void functions? Give examples.
10. Explain the working of python user defined functions along with its syntax.
11. List the uses of functions in Python.

Module 2 (Questions on Files Excluded):

1. Discuss looping constructs in Python.
2. In what situations, break and continue statements were used? Discuss with examples.
3. Explain different types of loop patterns in Python with suitable code.
4. How do you compare two strings in Python? Illustrate with example.
5. Discuss various built-in string functions.
6. Explain `.format()` with suitable examples.

Programming Questions:

1. Design a Python program to find the average of best two marks out of three marks taken as input.
2. A string with parentheses is well bracketed if all parentheses are matched: every opening bracket has a matching closing bracket and vice versa. Write a Python function `wellbracketed(s)` that takes a string `s` containing parentheses and returns True if `s` is well bracketed and False otherwise.

Here are some examples to show how your function should work.

```
>>> wellbracketed("22")
False
```

```
>>> wellbracketed("(a+b)(a-b)")
True
>>> wellbracketed("(a(b+c)-d)((e+f)")
False
```

3. A positive integer m is a sum of squares if it can be written as $k + x$ where $k > 0$, $x > 0$ and both k and x are perfect squares. Write a Python function *sumofsquares(m)* that takes an integer m returns True if m is a sum of squares and False otherwise. Here are some examples to show how your function should work.

```
>>> sumofsquares(41)
True
>>> sumofsquares(30)
False
>>> sumofsquares(17)
True
```

4. Write a program to reverse a string.
5. Write a program to display all the palindrome words appearing in an input text.
6. Write a program to list out all prime numbers between two integers m and n .
7. Write a program to count the frequency of characters in a string.
8. Write a Python function to get a string made of the first 2 and the last 2 chars from a given string. If the string length is less than 2, return empty string, otherwise return new string created.
9. Consider a string s ="Hello how are you?". Write a program to display this string without spaces.
10. Write a Python program to remove the characters which have odd index values of a given string.
11. Write a Python program to change a given string to a new string where the first and last chars have been exchanged.
12. Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged

```
Sample String : 'abc'
Expected Result : 'abcing'
Sample String : 'string'
Expected Result : 'stringly'
```