Python with Django QUESTION BANK

Input/Output:

1. Explain with syntax the use of input and output statements in python.
2. EXPLAIN WITH PYTHON CODE HOW, to input your name and print a welcome message along with your name.
3. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of replacement operators {}, {index} and {alphabet} in format() used in formatted string.
4. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of "%d", "%f", "%s" in a formatted string to print values of specific data types.
5. EXPLAIN WITH PYTHON CODE HOW, to find the sum of 3 numbers taking input from user. Print the 3 numbers and their sum as formatted string in the print() function.
6. EXPLAIN WITH PYTHON CODE HOW, to find the average if 3 numbers taking input from user. Print the 3 numbers and their average as formatted string in the print() function.
7. EXPLAIN WITH PYTHON CODE HOW, to sExplain with python code how, two variables. Take necessary inputs from user.
8. EXPLAIN WITH PYTHON CODE HOW, to sExplain with python code how, two variables without using a third variable. Take necessary inputs from user. (hint: use comma in between)
9. EXPLAIN WITH PYTHON CODE HOW, to use eval() to evaluate an arithmetic expression as a string input from user.
10. EXPLAIN WITH PYTHON CODE HOW, to calculate area of rectangle, square, circle and triangle. Take necessary inputs from user.
11. EXPLAIN WITH PYTHON CODE HOW, for height taken in cms then covert into feet and inches.(1 foot=12 inches and 1 inch=2.54 cm)
12. Accept as input the basic salary of a person. His dearness allowance (DA) is 40% of the basic salary and the house rent allowance (HRA) is 20% of the basic salary. Calculate and show the Gross salary.
13. Accept as Input the marks obtained by a student in 5 subjects. Show the Aggregate marks and Percentage marks.

Flow Control:

14. Explain with syntax the use of flow control statements in python.
15. EXPLAIN WITH PYTHON CODE HOW, to read age from keyboard and print whether the person is child,adult or elderly.
16. EXPLAIN WITH PYTHON CODE HOW, to find whether a given no is even or odd.
17. EXPLAIN WITH PYTHON CODE HOW, to find whether a given number is -ve , +ve or zero.
18. EXPLAIN WITH PYTHON CODE HOW, to find the greatest of 2 numbers taking input from user.
19. EXPLAIN WITH PYTHON CODE HOW, to find the smallest of 3 numbers taking input from user.
20. EXPLAIN WITH PYTHON CODE HOW, to find whether an input number is prime or composite.
21. EXPLAIN WITH PYTHON CODE HOW, in python to print whether a given character is an uppercase /lower case /a digit or any other character.

Python with Django Question Bank by Dr. Partha Roy

22. Accept a five digit number and reverse the number. Show whether the reversed number is same as the original number or not.
23. Accept the Cost price and Selling price of an item. Display whether there is profit or loss. Also determine how much profit is gained or loss is incurred.
24. Accept Basic salary and print the Gross salary according to the following conditions :-
    a) If Basic less than Rs1500.00 then HRA=10% and DA=90% of the basic salary.
    b) If Basic salary greater than or equal to Rs1500.00 then HRA=Rs500.00 and DA=98% of the basic salary.
25. Accept marks obtained by a student in 5 subjects. Print Division according to the following conditions :-
    a) Percentage above or equal to 60 – First division.
    b) Percentage between 50 and 59 – Second division.
    c) Percentage between 40 and 49 – Third division.
    d) Percentage below 40 - Fail
26. Accept two numbers and display the value of first number raised to the second number.
27. EXPLAIN WITH PYTHON CODE HOW, to print 1 to N integers in ascending order using while loop. Input N from user.
28. EXPLAIN WITH PYTHON CODE HOW, to print 1 to N integers in descending order using while loop. Input N from user.
29. EXPLAIN WITH PYTHON CODE HOW, to print sum of integers from 1 to N. Input N from user.
30. EXPLAIN WITH PYTHON CODE HOW, to print average of integers from 1 to N. Input N from user.
31. EXPLAIN WITH PYTHON CODE HOW, to print a string N number of times using repetition operator '*'. Input N from user.
32. EXPLAIN WITH PYTHON CODE HOW, to print a string N number of times using range(). Input N from user.
33. EXPLAIN WITH PYTHON CODE HOW, to print every individual character from a String input from user.
34. EXPLAIN WITH PYTHON CODE HOW, to print the index value of the characters present in the String input from user.
35. EXPLAIN WITH PYTHON CODE HOW, to print all the even numbers from 1 to N using range(). Input N from user.
36. EXPLAIN WITH PYTHON CODE HOW, to print all the odd numbers from 1 to N using range(). Input N from user.
37. EXPLAIN WITH PYTHON CODE HOW, to print all the even numbers from 1 to N using '%' operator. Input N from user.
38. EXPLAIN WITH PYTHON CODE HOW, to print all the odd numbers from 1 to N using '%' operator. Input N from user.
39. EXPLAIN WITH PYTHON CODE HOW, to print 10 to 1 in descending order using range().
40. EXPLAIN WITH PYTHON CODE HOW, to print N to 1 in descending order using range().Input N from user.
41. EXPLAIN WITH PYTHON CODE HOW, to read numbers in the form of list from user using eval(). Print the list and its data-type.
42. EXPLAIN WITH PYTHON CODE HOW, to read numbers in the form of tuple from user using eval(). Print the tuple and its data-type.
43. EXPLAIN WITH PYTHON CODE HOW, to read an integer, a float value and a string value using eval(). Print the values, their addresses and their data-types.

44. EXPLAIN WITH PYTHON CODE HOW, to print the Factorial of any number input from user.
45. EXPLAIN WITH PYTHON CODE HOW, to print the Fibonacci sequence up to N terms. Input N from user.
46. EXPLAIN WITH PYTHON CODE HOW, to print the following using nested while loop:
    * * * * *
    * * * *
    * * *
    * *
    *
47. EXPLAIN WITH PYTHON CODE HOW, to print the following using nested for loop:
    *
    * *
    * * *
    * * * *
    * * * * *
48. Write a program to print the following pattern .
    1
    12
    123
    1234
49. EXPLAIN WITH PYTHON CODE HOW, to input 5 numbers from the user. Exit the program and display "program terminated at 0" if user inputs 0, otherwise print at last "program completed successfully".
50. EXPLAIN WITH PYTHON CODE HOW, to print integers from 1 to N, the output should skip 2 and 7. Input N from user.

Strings:

51. Explain with syntax the use of Strings in python.
52. EXPLAIN WITH PYTHON CODE HOW, to print the Welcome message only when the input string is "3CSE" otherwise ask again for entering the required string. As long as "3CSE" is entered the program should keep asking the user.
53. EXPLAIN WITH PYTHON CODE HOW, to assume that uname="ABC" and pswd="123". Ask user to enter the correct combination of uname and pswd. Print "Welcome to Python" only when both the uname and pswd are correct, otherwise keep on asking user to enter correct uname and pswd.
54. EXPLAIN WITH PYTHON CODE HOW, to input a String from user and print the positive as well as negative index of each character present in the String.
55. EXPLAIN WITH PYTHON CODE HOW, to input start-index and end-index. Perform slicing operation on the following string "Welcome to Python".
56. EXPLAIN WITH PYTHON CODE HOW, using slicing operator print separately the characters present in even positions and odd positions in a String. Input the string from user.
57. EXPLAIN WITH PYTHON CODE HOW, to input a String from user and print that String in forward and reverse order using slicing operation. (hint: utilize the step-value)
58. EXPLAIN WITH PYTHON CODE HOW, to find whether the input String is a palindrome or not.

59. EXPLAIN WITH PYTHON CODE HOW, to print the number of alphabets, numbers and special symbols present in the String input from user.
60. EXPLAIN WITH PYTHON CODE HOW, to print the number of vowels present in the String input from user.
61. EXPLAIN WITH PYTHON CODE HOW, to check whether a substring is present in the main string, input both from the user. Use membership operator.
62. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of lstrip(), rstrip(), strip(), count(sub_string), count(sub_string,start_index,end_index) and replace(old_string,new_string) on Strings.
63. EXPLAIN WITH PYTHON CODE HOW, to prove that String objects are immutable.
64. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of count(), replace(), split(), rsplit(), join() on Strings.
65. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of upper(), sExplain with python code how, case(), title(), capitalize(), startswith(), endswith() on Strings.
66. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of isalpha(), isnumeric(), sorted(), chr() and ord() on Strings.

List:

67. Explain with syntax the application of lists in python.
68. EXPLAIN WITH PYTHON CODE HOW, to read a string from user and convert it into a list.
69. EXPLAIN WITH PYTHON CODE HOW, to demonstrate how to create a list 1) with no elements, 2) with elements by using String class split() and 3) with elements by using eval().
70. EXPLAIN WITH PYTHON CODE HOW, to demonstrate how to traverse a list 1) using print(), 2) using while loop and 3) using for loop.
71. EXPLAIN WITH PYTHON CODE HOW, to sExplain with python code how, two lists.
72. EXPLAIN WITH PYTHON CODE HOW, to print only the even numbers present in a List of integers.
73. EXPLAIN WITH PYTHON CODE HOW, to print the List elements along with its positive and its equivalent negative index.
74. EXPLAIN WITH PYTHON CODE HOW, to print the sum and average of all the elements in a List of numbers.
75. EXPLAIN WITH PYTHON CODE HOW, to print the greatest and smallest element present in a List of numbers.
76. EXPLAIN WITH PYTHON CODE HOW, to print the min and max values present in a homogeneous List.
77. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of append(), clear(), copy() and count() methods.
78. EXPLAIN WITH PYTHON CODE HOW, to create two Lists, the first List should contain only even numbers and second List should only contain odd numbers from a single main List of numbers.
79. EXPLAIN WITH PYTHON CODE HOW, to create a List of numbers from 1 to 100, where each element should be completely divisible by 10. (hint: use comprehension)
80. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of extend(), index() and insert() methods of List class.
81. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of pop(), remove() and reverse() methods of List class.

82. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of sort() method of List class, without and with 'reverse' and 'key'
83. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of sort() of List class, to sort the elements of the List containing Strings based on string length. (Hint: use 'key' parameter)
84. EXPLAIN WITH PYTHON CODE HOW, to print the nested list l1=[[10,20,30],[40,50,60],[70,80,90]] row-wise form and matrix form.
85. EXPLAIN WITH PYTHON CODE HOW, to find the greatest list from the following nested list l1=[[10,20,30],[40,50,60],[70,80,90]]
86. EXPLAIN WITH PYTHON CODE HOW, to create a list whose elements are squares of integers from 1 to 10. 1) Implement without list comprehension and 2) Implement with list comprehension.
87. EXPLAIN WITH PYTHON CODE HOW, to create a list whose elements are squares of integers from 1 to 20 and then create another list that contains only even numbers from the previous list and also create another list that contains only odd numbers from the previous list. Use List comprehension.
88. EXPLAIN WITH PYTHON CODE HOW, to build a list L1 containing ages ranging from 10 to 70 with a gap of 5 years in between. Extract a list L2 containing ages below 30. Extract another list L3 containing ages above 30 but below 50 and extract another list L4 having ages above 50. Use List comprehension.
89. EXPLAIN WITH PYTHON CODE HOW, to create a list L1 of integers from 1 to 10. From L1 create another list L2 with the condition that at positions having even numbers in L1 that even number will be inserted in L2 else 0 will be inserted. Use list comprehension.
90. EXPLAIN WITH PYTHON CODE HOW, to create a list containing only the first letter from each string present in another list of strings. Use list comprehension.
91. EXPLAIN WITH PYTHON CODE HOW, to create a list containing only the String length from each string present in another list of strings. Use list comprehension.
92. EXPLAIN WITH PYTHON CODE HOW, to create a list containing only those strings whose length is greater than 4 characters from each string present in another list of strings. Use list comprehension.
93. EXPLAIN WITH PYTHON CODE HOW, to create a list by extracting only those elements from first list that are not present in the second list. Use list comprehension.
94. EXPLAIN WITH PYTHON CODE HOW, to convert the following string "the quick brown fox jumps over the lazy dog" into list of words and then create another list that contains tuples having words and their respective lengths. Use list comprehension.
95. EXPLAIN WITH PYTHON CODE HOW, to convert the following string "the quick brown fox jumps over the lazy dog" into list of words and then create another list that contains uppercase words extracted from the previous list. Create another list that contains sub-list having uppercase words and its length. Use list comprehension.
96. EXPLAIN WITH PYTHON CODE HOW, to remove duplicate elements from a list and create a new list with those unique elements. (Hint: use in and not in)
97. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of append(object), clear(), copy(), count() and extend() methods on List object.
98. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of index(), insert(), pop(), remove(), reverse() and sort() methods on List object.

Python with Django Question Bank by Dr. Partha Roy

Tuple:

99. Explain with syntax the application of tuple in python.
100. EXPLAIN WITH PYTHON CODE HOW, to read a string from user and convert it into a tuple.
101. EXPLAIN WITH PYTHON CODE HOW, to demonstrate how to create a Tuple 1) with no elements, 2) with elements by using String class split() and 3) with elements by using eval().
102. EXPLAIN WITH PYTHON CODE HOW, to demonstrate how to traverse a Tuple 1) using print(), 2) using while loop and 3) using for loop.
103. EXPLAIN WITH PYTHON CODE HOW, to print only the even numbers present in a Tuple of integers.
104. EXPLAIN WITH PYTHON CODE HOW, to print the Tuple elements along with its positive and its equivalent negative index.
105. EXPLAIN WITH PYTHON CODE HOW, to print the sum and average of all the elements in a Tuple of numbers.
106. EXPLAIN WITH PYTHON CODE HOW, to print the greatest and smallest element present in a Tuple of numbers.
107. EXPLAIN WITH PYTHON CODE HOW, to print the min and max values present in a homogeneous Tuple.
108. EXPLAIN WITH PYTHON CODE HOW, to reverse a tuple
109. EXPLAIN WITH PYTHON CODE HOW, to print the value 20 from following tuple: t1 = ("Orange", [10, 20, 30], (5, 15, 25))
110. EXPLAIN WITH PYTHON CODE HOW, to create a tuple with single item 50
111. EXPLAIN WITH PYTHON CODE HOW, to create a tuple with 4 elements and upack them in 4 variables and print them
112. EXPLAIN WITH PYTHON CODE HOW, to sExplain with python code how, two tuples
113. EXPLAIN WITH PYTHON CODE HOW, to create a tuple t2 from an existing tuple t1 = (11, 22, 33, 44, 55, 66) by extrating 33,44 & 55 using slicing
114. EXPLAIN WITH PYTHON CODE HOW, to modify the first item (22) of a list inside a following tuple to 222 where t1=(11, [22, 33], 44, 55)
115. EXPLAIN WITH PYTHON CODE HOW, to create two Tuples, the first Tuple should contain only even numbers and second Tuple should only contain odd numbers from a single main Tuple of numbers. Use Tuple comprehension.
116. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the use of count() and index() methods on Tuple object.
117. EXPLAIN WITH PYTHON CODE HOW, to build a tuple T1 containing ages ranging from 10 to 70 with a gap of 5 years in between. Extract a tuple T2 containing ages below 30. Extract another tuple T3 containing ages above 30 but below 50 and extract another tuple T4 having ages above 50. Use Tuple comprehension.

Set:

118. Explain with syntax the application of set in python.
119. EXPLAIN WITH PYTHON CODE HOW, to read a string from user and convert it into a set.
120. EXPLAIN WITH PYTHON CODE HOW, to convert a list of numbers with duplicate entries into a set and display the result.
121. EXPLAIN WITH PYTHON CODE HOW, to create a set containing unique elements from another list. Use set comprehension.
122. EXPLAIN WITH PYTHON CODE HOW, to create a set containing integers from 10 to 100 with a gap of 10 between each element, using set comprehension.
123. EXPLAIN WITH PYTHON CODE HOW, to create a set containing even numbers from 1 to 10, using set comprehension.
124. EXPLAIN WITH PYTHON CODE HOW, to create a set containing even numbers from 1 to 10 and insert 0 if odd numbers are encountered. Use set comprehension.
125. EXPLAIN WITH PYTHON CODE HOW, to input values from user into two sets and perform set union, intersection and difference.
126. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of add(), clear(), copy(), difference(), discard(), intersection() and issubset() methods on Set objects.
127. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of issuperset(), pop(), remove(), symmetric_difference(), union() and update() on Set objects.

Dictionary:

128. Explain with syntax the application of dictionary in python.
129. EXPLAIN WITH PYTHON CODE HOW, to build a dictionary from two lists, one containing names of 5 usernames and another list containing their passwords. Use dictionary comprehension. (Hint: use zip())
130. EXPLAIN WITH PYTHON CODE HOW, to create two dictionaries, one should contain keys as even numbers and their respective values and another dictionary should contain as odd numbers and their respective values.List 1 contents are [1,2,3,4,5,6,7,8,9]. List 2 contents are extracted from the string "One Two Three Four Five Six Seven Eight Nine". Use Dictionary comprehension (Hint: use zip())
131. EXPLAIN WITH PYTHON CODE HOW, to read five subject names and their corresponding marks and store it in a dictionary. Display that dictionary.
132. EXPLAIN WITH PYTHON CODE HOW, to read five subject names and their corresponding marks and store it in a dictionary. Display subject name with marks that is maximum and minimum.
133. EXPLAIN WITH PYTHON CODE HOW, to create a Login validator. Use Dictionary to hold all users and their passwords. The existing users should be able to login by entering correct username and password.
134. EXPLAIN WITH PYTHON CODE HOW, to create a Login validator. Use Dictionary to hold all user names, their passwords and a secret key. The existing users should be able to login by entering correct username and password. Also the existing users can view and update their passwords by entering the right secret key .

135. EXPLAIN WITH PYTHON CODE HOW, to build a Dictionary to hold Name, Dept, Salary, DA and Gross of an employee.
    i. Input the Name, Dept and Salary details from the user.
    ii. Calculate DA as 20% of the Salary
    iii. Gross = Salary + DA
    iv. Display all the contents
136. EXPLAIN WITH PYTHON CODE HOW, to build a Dictionary to hold Name, Dept, AggMarks, AggPer and Div for a student.
    v. Input the Name and Dept details from the user.
    vi. Input marks of 5 subjects (out of 100) and store the aggregate in AggMarks.
    vii. Calculate Aggregate percentage out of 500 and store in AggMarks.
    viii. Display all the contents
137. EXPLAIN WITH PYTHON CODE HOW, to build a Dictionary to hold Name, Email and Address.
    ix. Input the Name, Email and Address.
    x. Provide facility to search any customer data using Email as the key.
    xi. Provide facility to delete any customer data using Email as the key.
    xii. Display all the contents
138. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of clear(), copy(), get(), items() and keys() methods on Dictionary object.
139. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of pop(), popitem(), update() and values() methods on Dictionary object.

General Purpose Methods:

140. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of following methods:
    i. abs(n)
    ii. all(iterable)
    iii. any(iterable)
    iv. chr(number)
    v. dir(object)
141. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of following methods:
    1. eval(string)
    2. help()
    3. id(object)
    4. input(message)
    5. len(object)
142. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of following methods:
    1. max(iterable)
    2. min(iterable)
    3. ord(character)
    4. pow(x,y)
    5. print(message)

143. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of following methods:
    1. reversed(iterable)
    2. round(n)
    3. sorted(iterable, key=key, reverse=reverse)
    4. sum(iterable)
    5. type(object)
    6. zip(iterable)

Class and Objects:

144. Explain with syntax the application of user defined class and objects in python.
145. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of class, objects, data members and member methods.
146. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of constructor.
147. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of destructor
148. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of method overriding.
149. EXPLAIN WITH PYTHON CODE HOW, to demonstrate two ways of calling the overridden parent class methods from child classes.
150. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the concept of instance members and static (class) members.
151. EXPLAIN WITH PYTHON CODE HOW, to print the unique ID of each object created and destroyed. Also print the number of objects created. (hint: use instance and static members)
152. EXPLAIN WITH PYTHON CODE HOW, to create a list of 5 objects and call the member methods of each object in the list.
153. EXPLAIN WITH PYTHON CODE HOW, to create a list of 5 students and each student should have its roll number, name and department. Implement necessary methods for filling and displaying the student details.
154. EXPLAIN WITH PYTHON CODE HOW, to create a list of 5 students and each student should have its roll number, name and department. Implement necessary methods for filling the student details, displaying the student details and also search a particular student detail by roll number.
155. EXPLAIN WITH PYTHON CODE HOW, to create a list of 5 students and each student should have its roll number, name and aggregate marks. Implement necessary methods for filling the student details, displaying the student details and also display the student with highest and lowest aggregate marks.
156. EXPLAIN WITH PYTHON CODE HOW, to demonstrate Single level inheritance.
157. EXPLAIN WITH PYTHON CODE HOW, to demonstrate Multi level inheritance.
158. EXPLAIN WITH PYTHON CODE HOW, to demonstrate Multiple level inheritance.
159. EXPLAIN WITH PYTHON CODE HOW, to demonstrate Hierarchical level inheritance.
160. EXPLAIN WITH PYTHON CODE HOW, to demonstrate Hybrid level inheritance.
161. EXPLAIN WITH PYTHON CODE HOW, to create a parent class named Shape with a member method area() which has no definition, derive two child classes Circle and Triangle, implement the area() in both the child classes and necessary member methods to input necessary data. Display the area of Circle

162. EXPLAIN WITH PYTHON CODE HOW, to create a parent class names Person with two member methods about() and biodata() without any definitions, derive two child classes named Actor and Actress, both should implement the about() to store the person's name along with other details and biodata() details of the cinema they have done. Also include necessary members so that we can store and view 5 Actor and Actresses details.
163. Define a Class that represents a bank account which has the following

    Data members :-
        Name of depositor.
        Account number.
        Balance amount.
    Member methods :-
        To assign initial values.
        To deposit an amount.
        To withdraw an amount after checking the balance.
        To display name and balance.

164. Define a Class that represents a bank account which has the following

    Data members :-
        Name of depositor.
        Account number.
        Balance amount.
    Member methods :-
        To assign initial values.
        To deposit an amount.
        To withdraw an amount after checking the balance.
        To display name and balance.
    Implement this for 5 customers.
165. Create a Class MyString, whose constructors can be used to perform following functions :-

        Create uninitialized strings
        Create objects initialized with a string
    Member methods :-
        To concatenate two strings and display the concatenated string.
        To display the contents of the objects.
166. Create a Class Book, which has following members :-

    Data members:
        Book title
        Name of the publisher
        Stock position of the books.
        Book price in Rupees
        Total sales in Rupees.
    Member methods:
        To Search for the book (using book title)
        To Sell the book and update the stock
        To Show the Stock position
        To show the Total Sales.

Exception Handling:

167. Explain with syntax the application of exception handling in python.
168. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of try, except, else and finally.
169. EXPLAIN WITH PYTHON CODE HOW, to demonstrate how to raise an inbuilt exception and a customized exception.
170. EXPLAIN WITH PYTHON CODE HOW, to generate a user defined exception whenever a numeric value is found in a string which is input from the user.
171. EXPLAIN WITH PYTHON CODE HOW, to generate a user defined exception whenever a character value is found in a string which is input from the user.
172. EXPLAIN WITH PYTHON CODE HOW, to generate a user defined exception whenever a special symbol is found in a string which is input from the user.
173. EXPLAIN WITH PYTHON CODE HOW, to input name and age from user. The program should generate an exception if name or age are not in proper format i.e. age should be numeric and name should contain only alphabets.
174. EXPLAIN WITH PYTHON CODE HOW, to input age from user. The program should generate an exception if age contains any non-numeric data or if the age entered is below 1 or above 90.
175. EXPLAIN WITH PYTHON CODE HOW, to develop a Name verifier, if the name contains any special symbol or numeric value then the program should generate a customized exception to indicate that only alphabets are allowed. The verification should be done using a dedicated method (eg. verifyName(String nm))

Database Programming:

176. Explain with syntax the application of CRUD in python.
177. EXPLAIN WITH PYTHON CODE HOW, to demonstrate CRUD operations.
178. EXPLAIN WITH PYTHON CODE HOW, to read student's name, five subject names and their corresponding marks and store it in a database table for 5 students. Display that table contents.
179. EXPLAIN WITH PYTHON CODE HOW, to read student's name, five subject names and their corresponding marks and store it in a database table for 5 students. Display subject names with maximum and minimum marks of every student.
180. EXPLAIN WITH PYTHON CODE HOW, to create a Login validator. Use database table to hold all users and their passwords. The existing users should be able to login by entering correct username and password.
181. EXPLAIN WITH PYTHON CODE HOW, to create a Login validator. Use database table to hold all user names, their passwords and a secret key. The existing users should be able to login by entering correct username and password. Also the existing users can view and update their passwords by entering the right secret key .
182. EXPLAIN WITH PYTHON CODE HOW, to build a database table to hold Name, Dept, Salary, DA and Gross of 5 employees.
    i. Input the Name, Dept and Salary details from the user.
    ii. Calculate DA as 20% of the Salary
    iii. Gross = Salary + DA
    iv. Display all the contents of the database table.

183. EXPLAIN WITH PYTHON CODE HOW, to build a database table to hold Name, Dept, Salary, DA and Gross of 5 employees.
    i. Input the Name, Dept and Salary details from the user.
    ii. Calculate DA as 20% of the Salary
    iii. Gross = Salary + DA.
    iv. Search any employee and display its details.
184. EXPLAIN WITH PYTHON CODE HOW, to build a database table to hold Name, Dept, AggMarks, AggPer and Div for 5 students.
    i. Input the Name and Dept details from the user.
    ii. Input marks of 5 subjects (out of 100) and store the aggregate in AggMarks.
    iii. Calculate Aggregate percentage out of 500 and store in AggMarks.
    iv. Display all the contents
185. EXPLAIN WITH PYTHON CODE HOW, to build a database table to hold Name, Email and Address of 5 customers.
    i. Input the Name, Email and Address.
    ii. Provide facility to search any customer data using Email as the key.
    iii. Provide facility to update any customer address using Email as the key.
    iv. Provide facility to delete any customer data using Email as the key.
    v. Display all the contents
186. Write a program to build a Student Information System. Use database table to store the data. The data base should contain the Enrollment Number, Student Name, Branch, Semester, Marks of 5 subjects. Give the facility to perform add, delete, search, edit and View All procedures on the database.
187. Create a database of 5 Customers in a Bank. Every customer has Account number, Name, Address of branch and Balance amount. Use database table to store the data. Provide following options to the user:
    1) Search any customer by Account number and display its details
    2) Edit any customer's address
    3) Deposit/Withdraw amount from the account.
    4) Display data of all the customers

Numpy:

188. Explain with syntax the application of Numpy and Random number generators in python.
189. Write a Python program to display all the elements present in the even locations in a Numpy array, build the array taking input from the user.
190. Write a Python program to add the elements present in the even locations with the elements present in the odd locations in a Numpy array, build the array taking input from the user.
191. Write a Python program to demonstrate how we can use tuple as well as list combined together to build a Numpy array, build the array taking input from the user.
192. Write a Python program to search any element input from user in a Numpy array. Build the array taking input from user.
193. Write a Python program to build a Numpy array of random integers. The range should be input from user.
194. Write a Python program to build a 3x3 Numpy array of random floating point numbers. The range should be input from user.

195. Write a Python program to build a 3x3 Numpy array of random integer numbers and multiply every element by any number input from user.
196. Write a Python program to build two 3x3 Numpy array of random integer numbers and add them, multiply them, divide them and get the modular division of them. Display each output separately.
197. Write a Python program to build one 3x3 Numpy array of random floating point values and then put any number input by user in both the diagonals of the martix and display it.
198. Find sum, product, mean, minimum, maximum, variance and standard deviation from the elements present in a Numpy array.
199. Demonstrate the concept of shallow copy and deep copy in Numpy arrays.
200. Write a Python program to build a 3x3 Numpy array of sequence of numbers from 1 to 9. Then use slicing to extract four 2x2 matrices from each corner and display all of them.
201. Write a Python program to build a 4x4 Numpy array of sequence of numbers from 1 to 16. Then use slicing to extract four 2x2 matrices without any overlapping values and display all of them.
202. Write a Python program to demonstrate horizontal and vertical stacking to Numpy arrays. Take input from user to build the arrays.
203. Write a Python program to demonstrate horizontal and vertical splitting to Numpy arrays. Take input from user to build the arrays.
204. Write a Python program to create a 3x3 Numpy array of random integers between 10 to 50. Filter out elements which are in the range of 10 to 20, 21 to 30, 31 to 40 and 41 to 50 into separate numpy arrays.
205. Write a Python program to create a 3x3 Numpy array of random integers between 1 to 10. Put 0 in place of all the elements which are in the range of 1 to 5 and put 1 in place of all the elements which are in the range of 6 to 10.
206. Write a Python program to create a 3x3 Numpy array of random integers between 10 to 20. Put False in place of all the elements which are in the range of 10 to 15 and put True in place of all the elements which are in the range of 16 to 20.
207. Write a Python program to create a 3x3 Numpy array of random integers from 10 to 50. Using where() function store even elements and odd elements in separate numpy arrays.
208. Write a Python program to read a line of text from user and convert it into a numpy array with words as its elements (treat this as our source array). Create one numpy array that contains all the words converted to upper case and another array that contains all the words converted to lower case from the source array.
209. Write a Python program to read a line of text from user and convert it into a numpy array with words as its elements (treat this as our source array). Create a numpy array that contains the string length of each element from the source array.
210. Write a Python program to read a line of text from user and convert it into a numpy array with words as its elements (treat this as our source array). Create a numpy array that contains the strings that contain only numbers in them from the source array.
211. Write a Python program to read a line of text from user and convert it into a numpy array with words as its elements (treat this as our source array). Create a numpy array that contains the sorted strings in ascending order from the source array.

212. Write a Python program to create a 3x3 Numpy array of random integers between 10 to 50 (treat this as our source array). Create atleast 3 arrays of different dimensions from the source array.
213. Write a Python program to create 3 Numpy array of random integers between 10 to 50. Use vertical stacking to build an array and horizontal stacking to create another array from the source arrays.
214. Write a Python program to create one Numpy array of random integers (treat this as our source array). Create three copies of the source array in such a way that if we perform changes in first and second copy then the all three including the source array also gets changed. And when we perform changes in the third copy then changes are not reflected in other copies including the source array.

Pandas:

215. Explain with syntax the application of Pandas series and data frame in python.
216. Write a Python program to create a pandas series from a list of elements. Create the list taking input from user.
217. Write a Python program to create a pandas series from a dictionary.
218. Write a Python program to create a pandas series from a list of elements. Customize the indexing of the series by taking input from user.
219. Write a Python program to create a pandas series from a list of 10 elements of person's ages randomly distributed between 18 to 90 years and extract all the elements that are at even index and odd index into two separate pandas series.
220. Write a Python program to create a pandas series from a list of elements and squareroot all the elements that are at even index and natural log the elements present in the odd index.
221. Write a Python program to create a pandas series from a list of 20 elements of person's ages randomly distributed between 18 to 90 years. Create three sub lists from the series having elements with range between 18 to 25, 26 to 45 and 46 to 90.
222. Write a Python program to demonstrate how dictionary can be used to build a dataframe.
223. Write a Python program to demonstrate how lists can be used to build a dataframe.
224. Write a Python program to create a pandas dataframe from a dictionary that contains keys as Email, Password, Age and Pincode. Prepare the dictionary to hold data for 5 values for every key. Create a dataframe from the dictionary with column names Email and Password.
225. Write a Python program to create a pandas dataframe with column names Email, Password, Age and Pincode. There should be 5 records. Using loc perform the following operations:
    1) Record present in row-index=2
    2) Record present in row-index=2 of column Email
    3) Records present from row-index 1 to 3
    4) Records present from row-index 1 to 3 of column Email and Pincode
    5) Records present at 0 and 3 row-index
    6) Extract individual column data into separate lists
    7) Display only those records where the Age is even
    8) Display only those records where the Pincode is odd
    9) Display only those records where the Age and Pincode both are even

226. Write a Python program to create a pandas dataframe from a dictionary. The column names should be Email, Password, Age and Pincode. The dataframe should contain 5 rows and each element should input from user.
227. Write a Python program to create a pandas dataframe having following column names: Email, Password, Hint Question,Hint Answer. Provide following facilties:
    1) New user registration.
    2) Existing user authentication.
    3) Password recovery on giving right hint answer.
228. Write a Python program to build a pandas dataframe of 5 customers having the following columns: Name, Email and Address.

    (i) Provide facility to search any customer data using Email as the key.
    (ii) Provide facility to delete any customer data using Email as the key.
    (iii) Display the contents.
229. Write a Python Program to build a pandas dataframe having following columns: EmpId, Name, Dept, Salary, DA and Gross of 5 employees. EmpId cannot be changed once created.

    (i) Input the Name, Dept and Salary details from the user.
    (ii) Calculate DA as 20% of the Salary
    (iii) Gross = Salary + DA
    (iv) Provide facility to search as well as update the Name/Dept/Salary.
230. Write a Python Program to build a pandas dataframe to hold RollNo, Name, Dept, AggMarks, AggPer and Div for storing information about 5 students. RollNo cannot be changed once created.

    (i) Input the Name and Dept details from the user.
    (ii) Input marks of 5 subjects and store the aggregate in AggMarks.
    (iii) Calculate Aggregate percentage out of 500 and store in AggMarks.
    (iv) Display the contents of the dataframe using an iterator.
231. Write a Python program to create a pandas dataframe from a csv file ('pulsedata.csv' from 'https://www.kaggle.com/drpartharoy/pulse-dataset-for-experiments') and perform the following operations:
232. Using iterator read and display individual column names.
233. Using iterator read and display individual row labels.
234. Using iterator read and display each row in the dataframe.
235. Write a Python program to create a pandas dataframe from a csv file ('pulsedata.csv' from 'https://www.kaggle.com/drpartharoy/pulse-dataset-for-experiments') and perform the following operations:
    1) Drop a single column.
    2) Drop mutliple columns.
    3) Drop a single row.
    4) Drop mutliple rows.
    5) Drop only those rows where all the row elements are even.

236. Write a Python program to create a pandas dataframe from a csv file ('pulsedata.csv' from 'https://www.kaggle.com/drpartharoy/pulse-dataset-for-experiments') and apply following statistical functions:
    1) count()
    2) sum()
    3) mean()
    4) median()
    5) mode()
    6) std()
    7) min()
    8) max()
    9) describe().
237. Write a Python program to create a pandas dataframe from a csv file ('pulsedata.csv' from 'https://www.kaggle.com/drpartharoy/pulse-dataset-for-experiments') and perform following operations:
    1) Sort the row labels.
    2) Sort the column labels.
    3) Sort data according to one column.
    4) Sort data according to multiple columns.
238. Write a Python program to create a pandas dataframe from a csv file ('pulsedata.csv' from 'https://www.kaggle.com/drpartharoy/pulse-dataset-for-experiments') and perform following operations:
    1) Find covariance of all the column data.
    2) Find correlation of all the column data.
    3) Find covariance between any two column data.
    4) Find correlation between any two column data.
239. Write a Python program to create a pandas dataframe from a csv file ('pulsedata.csv' from 'https://www.kaggle.com/drpartharoy/pulse-dataset-for-experiments') and perform the following operations using 'loc':
    1) Display all the even rows.
    2) Display all the odd rows.
    3) Display all rows from a 2 to 7.
    4) Display all the rows from first two columns.
    5) Display only those rows where 'Bsugar' is greater than 150.
    6) Display only those rows where 'Maxpulse' is between 130 and 150 .
    7) Display only those rows where 'Bsugar' is less than 150 and 'Maxpulse' greater than 150.
240. Write a Python program to create a pandas dataframe from a csv file ('pulsedata.csv' from 'https://www.kaggle.com/drpartharoy/pulse-dataset-for-experiments') and perform the following operations:
    1) Create an extra column that has square root of the square of the average of 'Pulse' and 'Bsugar' and name the new column as 'Indicator'.
    2) Create an extra column named 'Diagnosis' which should contain 1 in rows where 'Indicator' value of greater than 130 otherwise 0.
241. Write a Python program to create a pandas dataframe from a csv file ('cleaningdataset.csv' from 'https://www.kaggle.com/drpartharoy/sample-dataset-for-cleaning) and perform the following operations:
    1) Handling Empty cells.

2) Handling Data in wrong format
3) Handling Wrong data
4) Handling Duplicates.

242. Write a Python program to perform following operations using dataframe and csv file:
1) Reading entire csv file
2) Making the third row elements as the column headers
3) Creating the dataframe with the specified column data
4) Creating the dataframe with the specified multiple columns data
5) Skipping number of rows while reading the csv
6) Skipping specific rows while reading the csv
7) Specifying new set of column names while reading the csv
8) Writing the dataframe into a csv file

243. Write a Python program to read the table name: "2020 Racial composition of the population of Minnesota" from webpage: "https://en.wikipedia.org/wiki/Minnesota" into a dataframe and perform the activities:
1) Display the metadata of the table.
2) Convert the 'Percentage' column data type from object to float64.
3) Display the rows containing the highest and lowest percentage values.

244. Write a Python program to read the xml file: "studentdata.xml" from webpage: "https://www.kaggle.com/drpartharoy/studentdata" into a dataframe and perform the activities:
1) Display the metadata of the table.
2) Display only those rows where age is even.
3) Display only those rows where grade is A.

245. Write a Python program to read the excel file: "ExpertDetails.xlsx" from webpage: "https://www.kaggle.com/drpartharoy/expertdetails" into a dataframe and perform the activities:
1) Display the metadata of the table.
2) Display only those rows where Department is "Computer Science Engineering".
3) Display only those rows where Department is not "Computer Science Engineering".
4) Display only those rows where the Topic column contents contain the word "Python".
5) Sort the entire table in descending order of Expert's Name.
6) Sort the entire table in ascending order of Expert's Mobile number.
7) Make the column "Sl. No." as the row index.

Python with Django Question Bank by Dr. Partha Roy

Matplotlib:

246. Explain with syntax the application of Matplotlib in python.
247. Write separate Python programs to demonstrate the use of following charts:
    1) Line charts with their various parameters.
    2) Subplots with their various parameters.
    3) Scatter plots with their various parameters.
    4) Bar graphs with their various parameters.
    5) Histogram charts with their various parameters.
    6) Pie charts with their various parameters.
248. Write a Python program to create a pandas dataframe from a csv file ("microsoft.csv" from "https://www.kaggle.com/drpartharoy/microsoft") and perform the following operations:
    1) Create line charts from all the columns in the same figure.
    2) Create line charts as subplots from all the columns in the same figure.
    3) Create Scatter plots for all the columns pairing two columns at a time.
249. Write a Python program to create a pandas dataframe from a csv file ("facebook.csv" from "https://www.kaggle.com/drpartharoy/facebook") and perform the following operations:
    1) Create a new column named "Dir" which should be populated by only three values "UP", "DN" and "NT". When Open price of a day is equal to Low price of that day then the "Dir" value should be "UP". When Open price of a day is equal to High price of that day then the "Dir" value should be "DN", other wise the value should be "NT". Use List comprehension to populate the values.
    2) Create a Bar graph where x-axis should be the values in "Dir" column and y-axis should be the frequencey count of the values present in the "Dir" column.

Django:

250. Explain the application of MVT.
251. Compare MVC and MVT.
252. Describe the working algorithm of Django.
253. Explain Webserver, HTTP, HTTP request and HTTP response.
254. Practice the following: 1)Installing Django, 2)urls.py, 3)views.py, 4)startserver command, 5)runserver command, 6) templates folder, 7) static folder and 8)media folder.
255. EXPLAIN WITH PYTHON CODE HOW, on how to create Dynamic routes.
256. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the applications of 1) template variables, 2)template for loop, 3)template if-else, 4)template if-elif-else.
257. EXPLAIN WITH PYTHON CODE HOW, to render HTML template as response.
258. EXPLAIN WITH PYTHON CODE HOW, to implement a html form to read two numbers in separate text boxes and print the sum as result in Django.
259. EXPLAIN WITH PYTHON CODE HOW, to demonstrate the application of Include and Extends Django template tags.
260. EXPLAIN WITH PYTHON CODE HOW, to implement a html form to read the name of person through a text box and print a welcome message along with the person's name in Django.
261. EXPLAIN WITH PYTHON CODE HOW, to implement a html form to read two numbers in separate text boxes and print the sum as result in Django.

Python with Django Question Bank by Dr. Partha Roy

262. EXPLAIN WITH PYTHON CODE HOW, to implement a html form to build a simple numeric calculator in Django.

263. EXPLAIN WITH PYTHON CODE HOW, to input number of rows and columns and build a HTML table dynamically.

264. EXPLAIN WITH PYTHON CODE HOW, to input the employee name, department and basic salary and display the gross salary following the conditions mentioned below:-

a) If Basic less than Rs5000.00 then HRA=10% and DA=90% of the basic salary.

b) If Basic salary greater than Rs5000.00 and less than or equal to Rs15000 then HRA=20% and DA=61% of the basic salary.

c) If Basic salary greater than Rs15000 then HRA=30% and DA=72% of the basic salary.

265. EXPLAIN WITH PYTHON CODE HOW, to input the Student's Name, Enrollment Number, Marks obtained in 5 subjects (max marks per subject is 100) and display the result. The result should show the aggregate percentage and division: Above 75% Honors, between 65 and 75% First division, between 55 and 65% Second division and below 55 Third division.