**Comprehensive Examination, BITS Pilani**

**Semester-I, 2021-22**

|  |  |
| --- | --- |
| **Course Name**: Programming for Analytics | **Course Code**: MPBA G507 |
| **Time**: 4:00-6:00pm, 23 December 2021 |  |

|  |
| --- |
| **Student Name**: |
| **Student Enrolment Number**: |
| **Signature**: |
| **Invigilator Signature:** |

**Instructions to the students**:

1. Exam is in multiple choice questions format in Part-A & Part-B, and match the following in Part C
2. The entire exam is for 25 marks, and breakdown for each part is as follows: **Part-A**: 20x0.5 =10; **Part-B**: 10x1 = 10; and **Part-C**: 0.5x10 =5
3. Answer all the questions in page number 1.

**Part-A**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q.No.1** | **Q.No.2** | **Q.No.3** | **Q.No.4** | **Q.No.5** |
| **B** | **A** | **D** | **D** | **C** |
| **Q.No.6** | **Q.No.7** | **Q.No.8** | **Q.No.9** | **Q.No.10** |
| **B** | **C** | **D** | **C** | **D** |
| **Q.No.11** | **Q.No.12** | **Q.No.13** | **Q.No.14** | **Q.No.15** |
| **B** | **C or B** | **C** | **D** | **A** |
| **Q.No.16** | **Q.No.17** | **Q.No.18** | **Q.No.19** | **Q.No.20** |
| **B** | **D** | **A & D** | **A** | **B** |

**Part-B**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q.No.21** | **Q.No.22** | **Q.No.23** | **Q.No.24** | **Q.No.25** |
| **C** | **B** | **D** | **C** | **C** |
| **Q.No.26** | **Q.No.27** | **Q.No.28** | **Q.No.29** | **Q.No.30** |
| **B** | **D** | **C** | **B** | **A** |

**Part-C**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q.No.31** | **Q.No.32** | **Q.No.33** | **Q.No.34** | **Q.No.35** |
| **H** | **J** | **I** | **D** | **A** |
| **Q.No.36** | **Q.No.37** | **Q.No.38** | **Q.No.39** | **Q.No.40** |
| **C** | **B** | **E** | **G** | **F** |

**Part-A [20x0.5 = 10 marks]**

1. Select the correct output of the following code-snippet:

**x = {'a': [1,2,[3,4], 5, 6], 'b': [[5,6], 7]}**

**y = x['b'][0][1]**

**print(y)**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | 7 | b) | 6 |
| c) | y has no value. The following error occurred: IndexError: list index out of range | d) | y has no value. The following error occurred: KeyError: ‘b’ |

1. Select the correct output of the following code-snippet:

**x = "Welcome to BITS Pilani!"**

**x.split()[0][1]**

**print(x)**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | Welcome to BITS Pilani! | b) | Welcome |
| c) | W | d) | e |

1. The following code snippet throws an error message shown below. Select an option that rectifies the error.

Code snippet below:

**a = (1, 2, 3, 4, 5)**

**b = (6, 7, 8, 9, 10, 11, 12)**

**print(list(itertools.zip\_longest(a, b)))**

Error Message below:

NameError Traceback (most recent call last)

[<ipython-input-13-aabee836f5b8>](https://localhost:8080/) in <module>()

**2** b = (6, 7, 8, 9, 10, 11, 12)

**3**

----> 4 print(list(itertools.zip\_longest(a, b)))

NameError: name 'itertools' is not defined

|  |  |  |  |
| --- | --- | --- | --- |
| a) | library(itertools) | b) | package itertools |
| c) | Import itertools | d) | import itertools |

1. What will be the correct output of the following code-snippet:

**def x(x):**

**x = 5**

**return x**

**x = x(10)**

**print(x)**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | 10 | b) | <function i at 0xb7489764> |
| c) | FunctionOverride: function i assigned not called | d) | 5 |

1. What is the output of the following code-snippet?

**x=y=z=0**

**while True:**

**x += 1**

**if x<100:**

**y += 1**

**break**

**if x<200:**

**z += 1**

**break**

**else**

**break**

**break**

**print("Value of x: ", x)**

**print("Value of y: ", y)**

**print("Value of z: ", z)**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | Value of x: 1  Value of y: 1  Value of z: 0 | b) | Value of x: 99  Value of y: 99  Value of z: 100 |
| c) | SyntaxError: invalid syntax | d) | Value of x: 99  Value of y: 99  Value of z: 199 |

1. What is the output of the following code-snippet?

**x=4**

**y=0**

**if(y):**

**x += 2**

**print(x)**

**else:**

**x -= 2**

**print(x)**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | 4 | b) | 2 |
| c) | x | d) | SyntaxError: invalid syntax |

1. What is the output of the following code-snippet?

**a = "123"**

**b = "456"**

**if ("1" in a) ^ ("4" in b):**

**print("RED")**

**elif ("2" in a): print("Blue")**

**elif ("7" in b): print("Yellow")**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | Red  Blue | b) | Red  Blue  Yellow |
| c) | Blue | d) | Red |

1. What does the “enumerate” function return when applied to a list?

.

|  |  |  |  |
| --- | --- | --- | --- |
| a) | The memory location of the list | b) | The total number of items in the list |
| c) | A list containing a single tuple, with the tuple containing all items in the original list | d) | An object of tuples, with each assigned to an iterative integer |

1. What is the output of the following code-snippet?

**x = "Welcome to Programming for Analytics using R & Python"**

**x[:22][:10:-1]**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | Programming for Analytics using R & Python | b) | Programming |
| c) | gnimmargorP | d) | nohtyP & R gnisu scitylanA rof gnimmargorP |

1. In the following code snippet, today’s date in string format should be converted into date format. Identify the correct format of date in the **date\_format** string

**import time**

**date\_str = '23/12/2021'**

**datetime\_value = time.strptime(date\_str, date\_format)**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | "%dd/%mm/%YY" | b) | "%d/%m/%y" |
| c) | "%ds/%mm/%y" | d) | "%d/%m/%Y" |

1. **train** and **test** are two Numpy arrays. Create a new Numpy array by name **full** in the format given in table below. Identify the correct syntax to achieve creation of **full** array

|  |  |  |
| --- | --- | --- |
| **test**  [1 2 3] | **train**  [[0 1 2]  [1 2 3]] | **Full**  [[1, 2, 3],  [0, 1, 2],  [1, 2, 3]] |

|  |  |  |  |
| --- | --- | --- | --- |
| a) | np.concatenate(train, test) | b) | np.vstack([train, test]) |
| c) | np.vstack(test, train) | d) | np.concatenate([train, test]) |

1. What is the output of the following code-snippet?

**sorted([ 15 , "3" , 10.0 , "4.0" ])**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | [“3.0”, “4.0”, “10.0”, “15.0”] | b) | [“3”, “4”, “10”, “15”] |
| c) | TypeError: '<' not supported between instances of 'str' and 'int' | d) | [3.0, 4.0, 10.0, 15.0] |

1. What is the output of the following code-snippet?

**class Greet :**

**msg = 'Hello'**

**g1 = Greet()**

**g2 = Greet()**

**g2.msg = 'Welcome'**

**Greet.msg = 'Hi'**

**print(g1.msg, g2.msg, Greet.msg)**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | Hello Welcome Hi | b) | Hello Welcome Hello |
| c) | Hi Welcome Hi | d) | Hi Welcome Hello |

1. What is the output of the following code-snippet?

**class FirstClass :**

**def \_\_init\_\_ (self):**

**self.var1 = 1**

**class SecondClass (FirstClass):**

**def \_\_init\_\_ (self):**

**self.var2 = 2**

**x = SecondClass()**

**print(x.var1, x.var2)**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | 2 1 | b) | 1 2 |
| c) | NaN 2 | d) | AttributeError: 'SecondClass' object has no attribute 'var1' |

1. What is the output of the following code-snippet?

**class FirstClass :**

**v = 1**

**def v0 (self):**

**return self.v**

**class SecondClass (FirstClass):**

**v = 2**

**s = SecondClass()**

**h = FirstClass()**

**print(s.v0(), h.v0())**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | 2 1 | b) | 2 2 |
| c) | 1 2 | d) | 1 1 |
| e) | AttributeError: 'SecondClass' object has no attribute 'v0()' | | |

1. Select the correct option in place of <<INSERT CODE HERE>> to run following code-snippet without any errors, and displays result of 1 2.

**class FirstClass :**

**def \_\_init\_\_ (self):**

**self.x = 1**

**class SecondClass (FirstClass):**

**def \_\_init\_\_(self):**

**<<INSERT CODE HERE>>**

**self.y = 2**

**f = SecondClass()**

**print(f.x, f.y)**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | super(Bar, self).\_\_init\_\_() | b) | FirstClass.\_\_init\_\_(self) |
| c) | self.x =2 | d) | Nothing needs to be inserted |

1. What is the output of the following code snippet?

**[i \* i for i in range( 1 , 3 )]**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | [1, 2, 3] | b) | [1, 4, 9] |
| c) | [1 , 2] | d) | [1, 4] |

**18**. Select one or more than one correct valid definition of a **lambda** assigned to **f** that adds the arguments **x** and **y**?

|  |  |  |  |
| --- | --- | --- | --- |
| a) | f = lambda x, y : x + y | b) | f = lambda (x, y):(x + y) |
| c) | f = lambda (x, y): x + y | d) | f = lambda x, y : (x + y) |

1. What is the output of the following code snippet?

**total = 0**

**for i in range( 1 , 3 ) :**

**i += 2**

**total += i**

**else :**

**total += 10**

**print(total)**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | 17 | b) | 7 |
| c) | SyntaxError: invalid syntax | d) | 117 |

**20**. **x** is a list of elements declared as **x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]**. Select all the options that **does not** result in subset of tuple **(0, 3, 6, 9)** from the list **x**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | x[::3] | b) | x[0:3:9] |
| c) | x[0::3] | d) | x[0:10:3] |

**Part-B [10x1 = 10 marks]**

**21**. Which command invokes method **message()** of the object **greetings**?  
Note: **greetings** is an object of a class that consists of **message()** method

|  |  |  |  |
| --- | --- | --- | --- |
| a) | message()greetings | b) | greetings$message() |
| c) | greetings.message() | d) | greetings(message) |

**22**. What is the output of the following code snippet?

**x = 6**

**def x\_square(x=5):**

**return x\*x**

**x\_square(x)**

**print(x)**

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| a) | 36 | b) | 6 |
| c) | 5 | d) | 25 |

**23**. What is the correct way of calling **function2()** independently?

**def function1():**

**def function2():**

**return 2**

**return 1**

|  |  |  |  |
| --- | --- | --- | --- |
| a) | function2() | b) | function1.function2() |
| c) | function1().function2() | d) | cannot call a function that is defined in another function |

Review the following pandas dataframe saved as **df**, and answer the questions from **24** to **30**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **EmpID** | **FirstName** | **Salary** | **Dept** | **DoJ** |
| 0 | E0657 | Vaibhav | 102,000 | IT | 11/12/2020 |
| 1 | E0256 | Rahul | 151,233 | DS | 01/05/2019 |
| 2 | E0542 | Saif | 96,440 | FIN | 06/12/2019 |
| 3 | E0124 | Megha | 99,531 | DS | NaN |
| 4 | E0896 | Rachel | 125,660 | IT | 01/02/2019 |
| 5 | E0633 | Mike | NaN | HR | 19/12/2019 |

**24**. Which of the following code fetches all the records with salary less than 1,00,000?

|  |  |  |  |
| --- | --- | --- | --- |
| a) | df.[df(Salary < 100000)] | b) | df.'Salary' < 100000 |
| c) | df[df.Salary < 100000] | d) | df.Salary < 100000 |

**25**. Which of the following code fetches all the records and arranges the FirstName in alphabetical order (beginning with A and ending with Z)

|  |  |  |  |
| --- | --- | --- | --- |
| a) | df.sort\_values(by='FirstName',  ascending = False) | b) | df['FirstName'].sort\_values() |
| c) | df.sort\_values('FirstName') | d) | df[:,2:].sort\_values() |

**26**. Which of the following code replaces the NaN from the Salary column with 104500?

|  |  |  |  |
| --- | --- | --- | --- |
| a) | df.replacena(104500) | b) | df['Salary'].fillna(104500) |
| c) | df['Salary'].replacena(104500) | d) | df.fillna(104500) |

**27**. Which of the following code adds a new column bonus which is 30% of the Salary

|  |  |  |  |
| --- | --- | --- | --- |
| a) | df['bonus'] = df['Salary']\*30 | b) | df['bonus' 'Salary'\*30] |
| c) | df['bonus' 'Salary'\*0.3] | d) | df['bonus'] = df['Salary']\*0.3 |

**28**. Which of the following code drops the column Dept?

|  |  |  |  |
| --- | --- | --- | --- |
| a) | df.drop['Dept'] | b) | df.drop(['Dept']) |
| c) | df.drop(['Dept'], axis=1) | d) | df.drop(['Dept'], axis=0) |

**29**. Which of the following code finds all employees joined in the month of December? Assume that the datatype of DoJ column is datetime64[ns

|  |  |  |  |
| --- | --- | --- | --- |
| a) | df['DoJ'].dt.month ==12 | b) | df[df['DoJ'].dt.month ==12] |
| c) | df[df['DoJ'].month ==12] | d) | df[df['DoJ'].month ==12] |

**30**. Which of the following line of code provides mean salary per Dept (department column)?

|  |  |  |  |
| --- | --- | --- | --- |
| a) | df.groupby('Dept')['Salary'].mean() | b) | df.groupby('Dept')['Salary'].np.mean() |
| c) | df['Salary'].groupby('Dept').mean() | d) | df['Salary'].groupby('Dept').np.mean() |

**Part-C [10x0.5 = 5 marks]**

Match the following methods/functions/attributes applied on Numpy Arrays or Pandas Dataframe or Series with Description column in the table below.

|  |  |  |
| --- | --- | --- |
| **Q.No.** | **Method/Function/Attribute** | **Description** |
| 31 | shape | 1. Count each unique value’s number of occurrences in pandas dataframe column |
| 32 | dtypes | 1. Join DataFrame or named Series objects with a database-style join. |
| 33 | iloc | 1. Join columns with other DataFrame either on index or on a key column without using database-style join. |
| 34 | loc | 1. Extracts a row by its index label rather than its numeric position |
| 35 | value\_counts() | 1. Combine or join pandas objects along a particular axis |
| 36 | join() | 1. Counts number of values in each column excluding NAs |
| 37 | merge() | 1. Return a subset of the DataFrame’s columns based on the column dtypes. |
| 38 | concat() | 1. The number of rows and columns in the DataFrame |
| 39 | select\_dtypes() | 1. Extract a row from the pandas DataFrame or Series (dataset) by its numeric order in line |
| 40 | count() | 1. Data types in columns of pandas DataFrame or Series, and numpy arrays |