CONFIDENTIAL

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name of Examination**  (Please tick, symbol is given) | **:** | **MID** |  | | | **END** |  | **SUPPLE** |  |
| **Name of the School**  (Please tick, symbol is given) | **:** | **SOE** |  | | | **SOCS** |  | **SOP** |  |
| **Programme** | **:** | **B.Tech IOT, ECRA, OGI, MC** | | | | | | | |
| **Semester** | **:** | **V** | | | | | | | |
| **Name of the Course** | **:** | **Interactive Programming through Python** | | | | | | | |
| **Course Code** | **:** | **CSEG 3017** | | | | | | | |
| **Name of Question Paper Setter** | **:** | **Deepa Joshi** | | | | | | | |
| **Employee Code** | **:** | **40001464** | | | | | | | |
| **Mobile & Extension** | **:** | **7060225471** | | | | | | | |
| **Note: Please mention additional Stationery to be provided, during examination such as Table/Graph Sheet etc. else mention “NOT APPLICABLE”:** | | | | | | | | | |
| **FOR SRE DEPARTMENT** | | | | | | | | | |
| **Date of Examination** | | | | **:** |  | | | | |
| **Time of Examination** | | | | **:** |  | | | | |
| **No. of Copies (for Print)** | | | | **:** |  | | | | |

**Note: - Pl. start your question paper from next page**

**Model Question Paper (Blank) is on next page**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name:**  **Enrolment No:** | |  | | |
| UNIVERSITY OF PETROLEUM AND ENERGY STUDIESEnd Semester Examination, December 2018Course: Interactive Programming through Python (CSEG 3017) Semester: V **Programme:** B.Tech (IOT,ECRA,OGI,MC)  **Time: 03 hrs. Max. Marks: 100**  **Instructions: Attempt all questions** | | | | |
| **SECTION A** | | | | |
| S. No. |  | | **Marks** | **CO** |
| Q1. | Explain use of **\*\*, /, //** and **in** operators in python [1+1+1+1]  Ans:   1. \*\* Performs exponential (power) calculation on operators. a\*\*b =10 to the power 20 2. / Divides left hand operand by right hand operand 20/10=2 3. // Floor Division In Python 3.0, 5 / 2 will return 2.5 and 5 // 2 will return 2. The former is floating point division, and the latter is floor division, sometimes also called integer division. It Evaluates to true if it finds a variable in the specified sequence such as strings, lists, or tuples and false otherwise.   a = 10  b = 20  list = [1, 2, 3, 4, 5 ];  if ( a in list ):  print "Line 1 - a is available in the given list"  else:  print "Line 1 - a is not available in the given list"  Output: Line 1 - a is not available in the given list | | **4** | **CO1** |
| Q2. | Describe various types of arguments used in the functions.  Ans: We can call a function by using the following types of formal arguments −   1. Positional or Required arguments: Required arguments are the arguments passed to a function in correct positional order. Here, the number of arguments in the function call should match exactly with the function definition.      1. Keyword arguments: When we use keyword arguments in a function call, the caller identifies the arguments by the parameter name. This allows us to skip arguments or place them out of order.      1. Default arguments: A default argument is an argument that assumes a default value if a value is not provided in the function call for that argument.      1. Variable-length arguments: To process a function for more arguments than the specified while defining the function.     An asterisk (\*) is placed before the variable name that holds the values of all nonkeyword variable arguments. This tuple remains empty if no additional arguments are specified during the function call | | **4** | **CO1** |
| Q3. | Why python does not supports overloading concept. Justify your answer with an example.  Ans: In Python you can define a method in such a way that there are multiple ways to call it. Given a single method or function, we can specify the number of parameters ourself. Depending on the function definition, it can be called with zero, one, two or more parameters.    python does not supports method overloading. We may overload the methods but can only use the latest defined method.  def product(a, b):      p = a \* b      print(p)  def product(a, b, c):      p = a \* b\*c      print(p)  product(4, 5) #line shows an error  product(4, 5, 5)  This line will call the second product method | | **4** | **CO2** |
| Q4. | **Explain** any **four** keywords used in Exception handling in python. [1+1+1+1]  Ans:1. try block: In the try block, we can write a section of code that could probably raise an error.  2. except block: The except block is then written so that if any exception comes true, the control of the program will be passed to the except block and we could thus prevent program from abnormal termination.  3. finally block: A finally block is very useful in Python exception handling. A finally clause always gets executed as soon the control completes the try block. It doesn’t matter whether an exception has occurred or not.  4. else block: If there is no any exception, then this block is executed | | **4** | **CO1** |
| Q5. | Find the output of the above program [2]  Ans:      **True/False.**   1. If the object passed to a function is of immutable type, the passing acts like pass by value. [1] True 2. If the object passed to a function is of mutable type, the passing acts like pass by reference. [1] True | | **4** | **CO2** |
| **SECTION B** | | | | |
| Q6. | Create database connectivity with python and perform the queries given below in a single file.   1. Create database connection [2] 2. Create a table named **student\_info** with the following fields (name, SAP\_ID, address, Date\_of\_birth) [2] 3. Insert records of three students into the **student\_info** table [2] 4. Select students whose name starts with H from **student\_info** table [2] 5. Sort the **name** field in descending order. [2]   Ans:  a.  import mysql.connector as msc  connection = msc.connect(  host="localhost",  user="root",  passwd="deepa",  database="ecra\_students"  )  cursor = connection.cursor()  cursor.execute ("SELECT VERSION()")  row = cursor.fetchone()  print("server version:", row[0])  b.  query=”””CREATE TABLE employee (  SAP\_ID INTEGER PRIMARY KEY,  fname VARCHAR(20),  lname VARCHAR(20),  Address VARCHAR(30),  birth\_date DATE);"""  cursor.execute(query)  print("Record created successfully")  c.  staff\_data = [ ("Ailliam", "Shakespeare", "address1", "1961-10-25"),  ("Hrank", "Schiller", "address2", "1955-08-17"),  ("Mane", "Wall", "address3", "1989-03-14"),  ]  for staff, p in enumerate(staff\_data):  format\_str = """INSERT INTO employee (SAP\_ID, fname, lname, address, birth\_date)  VALUES ({SAP\_ID}, '{first}', '{last}', '{address}', '{birthdate}');"""  query = format\_str.format(SAP\_ID=staff, first=p[0], last=p[1], address=p[2], birthdate = p[3])  cursor.execute(query)  print("data inserted successfully")  d.  cursor.execute("SELECT fname FROM employee where fname LIKE 'H%'")  print('''Result of "SELECT \* FROM employee":''')  result = cursor.fetchall()  print("total result is\n\n", result)  print("\n")  for r in result:  print(r)  e.  cursor.execute("SELECT \* FROM employee ORDER BY fname DESC")  result = cursor.fetchall()  print("total result is\n\n", result)  print("\n")  for r in result:  print(r)  # never forget this, if you want the changes to be saved:  connection.commit()  connection.close() | | **10** | **CO4** |
| Q7. | Assume a file city.txt with details of 5 cities in given format (cityname  population(in lakhs) area(in sq KM) ):  Example:  Dehradun 5.78 308.20  Delhi 190 1484  ……………  Open file city.txt and read to:   1. Display details of all cities [3] 2. Display city names with population more than 10Lakhs [4] 3. Display sum of areas of all cities [3] | | **10** | **CO3** |
| Q8. | Answer following questions:   1. Convert **numbers =[1, 2.0, 3]** to numpy array and convert all elements to string type. [2]   **import numpy as np**  **numbers=np.array([1, 2.0, 3], dtype=str)**   1. Create a 2 D array through list and set dtype as int32 [2]   **numbers=np.array([[1, 2, 3],[4,5,6,]], dtype=np.int32)**   1. Find the rows and columns of the 2d array created in part b [2]      1. Write the output of: np.arange(16).reshape(2,2,4) [2]      1. Write the output of : np.random.randint(1,100,10) [2]   Ans: It will print 10 random numbers between 1 and 100. Example    **OR**  Refer the code and find the output:     1. Find a & b from the above code. [2]      1. Perform a+b, a-b, a\*b and find the output. [2]      1. Find output of a<35 [2]      1. Which method is used to perform matrix multiplication using numpy? [2]   numpy.dot   1. If a= [[0, 1, 2, 3],   [4, 5, 6, 7],  [8, 9, 10, 11]]    Find a.min(axis=1) [2] | | **10** | **CO5** |
| Q9. | Take one example and write python code to discuss the importance of synchronization in threads.  import threading    # global variable x  x = 0    def increment():  """  function to increment global variable x  """  global x  x += 1    def thread\_task(lock):  """  task for thread  calls increment function 100000 times.  """  for \_ in range(100000):  lock.acquire()  increment()  lock.release()    def main\_task():  global x  # setting global variable x as 0  x = 0    # creating a lock  lock = threading.Lock()    # creating threads  t1 = threading.Thread(target=thread\_task, args=(lock,))  t2 = threading.Thread(target=thread\_task, args=(lock,))    # start threads  t1.start()  t2.start()    # wait until threads finish their job  t1.join()  t2.join()    if \_\_name\_\_ == "\_\_main\_\_":  for i in range(10):  main\_task()  print("Iteration {0}: x = {1}".format(i,x))  Output:  Iteration 0: x = 200000  Iteration 1: x = 200000  Iteration 2: x = 200000  Iteration 3: x = 200000  Iteration 4: x = 200000  Iteration 5: x = 200000  Iteration 6: x = 200000  Iteration 7: x = 200000  Iteration 8: x = 200000  Iteration 9: x = 200000 | | **10** | **CO1** |
| **SECTION-C** | | | | |
| Q10. | Refer given csv file and answer given questions: | |  | **CO5** |
|  | 1. Import the given csv file using pandas. (File name is weather.csv)   **import pandas as pd**  **temp=pd.read\_csv(weather.csv')**   1. Find maximum temperature.   **df[‘Temperature’].max()**   1. Find average WindSpeed.   **df[‘WindSpeedMPH’].mean()**   1. Retrieve Date when the **Events** was “**rain**”   **df[‘EST’][df[‘Events’]==’Rain’]**   1. Find number of rows and columns present in the file.   **df.shape**   1. Print **Humidity** and **Events** columns from the file.   **df[[’Humidity’, ‘Events’]]**   1. Find all the rows where temperature is greater than 32.   **df[df[‘Temperature’]>32]**   1. Change the index to date on which temperature recorded.   **df.set\_index(‘EST’, inplace=’True’)**   1. Print the **temperature** and **day** on which the temperature was **maximum.**   **df[[‘Day’, ‘Temperature’]]**  **[df[‘Temperature’]==df[‘Temperature’].max()**]   1. Fill NAN values present in the **temperature** column with **0** and fill NAN value present in **Events** column with “**no event**”.   **df.fillna({‘Temperature:0’,**  **Events:’no event’})** | | **20** |  |
| Q11. | 1. Write a function, which takes list of integer values, and returns cube of each element in list.   **e.g. input=[2,3,5,1,4]     expected output: [8,27,125,1,64]** [2]  **numbers=[2,3,5,1,4]**  **squares=list(map(lambda x:pow(x,2),numbers))**  **print(squares)**  **or**  **def turn\_to\_power(list, power=1):**  **return [number\*\*power for number in list]**  b. Can you store the details of multiple books in a dictionary at the same time? Details include- book\_id, book\_name , price & year . Give example to support your answer. [4]  c. Suggest suitable structure for storing the above details of 5 books and print book\_name & year, number of books with price>=150. [7]  d. Write a function, which takes structure, taken in answer of (c), as argument and returns total price of all books. [7]  **OR**  The "Variety Retail Store" sells diﬀerent varieties of Furniture to the customers. The list of furniture available with its respective cost is given below:    The furniture and its corresponding cost should be stored as a list. A customer can order any furniture in any quantity (the name and quantity of the furniture will be provided). If the required furniture is available in the furniture list (given above) and quantity to be purchased is greater than zero, then bill amount should be calculated. In case of invalid values for furniture required by the customer and quantity to be purchased, display appropriate error message and consider bill amount to be 0. Initialize required furniture and quantity with diﬀerent values and test the results. Write a Python program to calculate and display the bill amount to be paid by the customer based on the furniture bought and quantity purchased. | | **20** | **CO2** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name:**  **Enrolment No:** | |  | | |
| UNIVERSITY OF PETROLEUM AND ENERGY STUDIESEnd Semester Examination, December 2018Course: Interactive Programming through Python (CSEG 3017 ) Semester: V **Programme:** B.Tech (IOT,ECRA,OGI,MC)  **Time: 03 hrs. Max. Marks: 100**  **Instructions: Attempt all questions** | | | | |
| **SECTION A** | | | | |
| S. No. |  | | **Marks** |  |
| Q1. | When the else part of try-except-else be executed? [2] Why the code given below will not compile, give justification. [2]  try:  print("Hello world!")  except:  print('Error occured')  except(TypeError):  print("Invalid Datatype")  except(ValueError):  print("Invalid Value")  finally:  print("Last block") | | **4** | **CO1** |
| Q2. | Differentiate mutable and immutable datatypes with help of examples. | | **4** | **CO2** |
| Q3. | Explain different types of inheritance in class with example. | | **4** | **CO1** |
| Q4. | Explain python lambda functions with the help of an example. | | **4** | **CO2** |
| Q5. | Explain the concept of overloading and overriding in python with the help of examples. [2+2] | | **4** | **CO1** |
| **SECTION B** | | | | |
| Q6. | Create database connectivity with python and perform the queries given below in a single file.   1. Create database connection. [2] 2. Create a table named **employee\_info** with the following fields (emp\_name, EMP\_ID, address, Date\_of\_joining and salary) [2] 3. Insert record of five employees into the table. [2] 4. Select employee whose salary is greater than 25000 from the **employee\_info** table. [2] 5. Select employee whose name starts with G and Date\_of\_joining after july 2018. [2] | | **10** | **CO4** |
| Q7. | 1. Create a file student.txt and insert details of 5 students in given format (student\_name  Roll\_no Marks)   **Example**:  Ram 10 72  Shyam 20 55  ……………  Ans:  with open('D://student\_info.txt','a') as file:  print(file.tell())  lines=['Ram 10 72\n','Shyam 20 87\n']  file.writelines(lines)  print("Record has been written to file")  [5]   1. Open file student.txt and find average marks of 5 students stored in the file. [5]   **Ans:** sum=0  with open('D://student\_info.txt','r') as file:  for line in file:  values = line.split()  sum=sum+int(values[2])  print(sum) | | **10** | **CO3** |
| Q8. | Refer the code and find the output:     1. Find a & b from the above code. [2] 2. Perform a+b, a-b, a\*b and find the output. [2] 3. Find output of a<35 [2] 4. Which method is used to perform matrix multiplication using numpy? [2] 5. If a= [[0, 1, 2, 3],   [4, 5, 6, 7],  [8, 9, 10, 11]]    Find a.min(axis=1) [2] | | **10** | **CO5** |
| Q9. | Take one example and write python code to discuss the importance of synchronization in threads.  OR  Write a program that takes a string with multiple words and then capitalizes the first letter of each word and forms a new string out of it. | | **10** | **CO1** |
| **SECTION-C** | | | | |
| Q10. | Refer the given excel file and perform various operations: | |  |  |
|  | 1. Read the above **excel** file in python. [2] 2. How do I write this file to a new file “**new.csv”?** [2] 3. Include column names in this file. Use **‘ticker’, ‘eps’, ‘revenue’, ‘price’, ‘people’** as column names. [3] 4. Convert all **not available** or **n.a.** values to NAN and also convert negative revenues to NAN because revenues can never be negative. [4] 5. Fill NAN values using a suitable approach. [4] 6. Write a function to change n.a value appearing in WMT to **Sam Walton**. [5] | | **20** | **CO5** |
| Q11. | Create four Employees with following properties:  First name, last name, employee code, monthly\_pay and email id where email id is a combination of firstname, lastname and xyz.com. [Example: firstname.lastname @xyz.com]  The ﬁnance department of a company wants to compute the monthly pay of its employees. Monthly pay is calculated as mentioned in the formula below.   * Monthly Pay = Number of hours worked in a week \* Pay rate per hour \* No. of weeks in a month * The number of hours worked by the employee in a week should be considered as 40 * Pay rate per hour should be considered as Rs.400 * Number of weeks in a month should be considered as 5   Create three methods display\_info() to display employee information, monthly\_pay() to display monthly income of employee, apply\_raise() to display the increased salary. Every year a raise is given to each employee in his/her salary. However, the raise amount varies every year but is equal for all the employees. Write a Python program to implement the above real world problem. Also, find total number of employees exists in the organization.  **OR**  Given below is a dictionary 'customer\_details' representing customer details from a Retail Application. Customer Id is the key and Customer Name is the value.  **customer\_details = { 1001 : "John", 1004 : "Jill", 1005: "Joe", 1003 : "Jack" }**  Write Python code to perform the operations mentioned below:  a) Print details of customers. [3]  b) Print number of customers. [3]  c) Print customer names in ascending order. [4]  d) Delete the details of customer with customer id = 1005 and print updated dictionary. [5]  e) Update the name of customer with customer id = 1003 to "Mary" and print updated dictionary. [5] | | **20** | **CO1** |