**In-Lab**

**In-Lab Task 1**

|  |  |
| --- | --- |
| **Code:**  phonebook = {}  phonebook["Afaan Kamran"]={"Phone":"0304-4657625", "Email":"fa20-bce-053@cuilahore.edu.pk"}  phonebook["Junaid Salman"]={"Phone":"0301-5369018", "Email":"fa20-bce-044@cuilahore.edu.pk"}  phonebook["Rana Fahad Aman"]={"Phone":"0349-4945272", "Email":"fa20-bce-021@cuilahore.edu.pk"}  print(phonebook)  **Output:**   |  | | --- | | {  'Afaan Kamran':  {'Phone': '0304-4657625', 'Email': 'fa20-bce-053@cuilahore.edu.pk'},  'Junaid Salman':  {'Phone': '0301-5369018', 'Email': 'fa20-bce-044@cuilahore.edu.pk'},  'Rana Fahad Aman':  {'Phone': '0349-4945272', 'Email': 'fa20-bce-021@cuilahore.edu.pk'}  } | |

**In-Lab Task 2**

|  |  |
| --- | --- |
| **Code:**  for name, record in phonebook.items():  print("{}'s phone number is {}, and email is {}".format(name,record["Phone"],record["Email"]))  **Output:**   |  | | --- | | Afaan Kamran's phone number is 0304-4657625, and email is [fa20-bce-053@cuilahore.edu.pk](mailto:fa20-bce-053@cuilahore.edu.pk)  Junaid Salman's phone number is 0301-5369018, and email is fa20-bce-044@cuilahore.edu.pk  Rana Fahad Aman's phone number is 0349-4945272, and email is fa20-bce-021@cuilahore.edu.pk | |

**In-Lab Task 3**

|  |  |
| --- | --- |
| **Code:**  for name, record in phonebook.items():  print("{}'s phone number is {}, and email is {}.".format(name, record["Phone"],record["Email"]))  print("\n")  del phonebook["Afaan Kamran"]  for name, record in phonebook.items():  print("{}'s phone number is {}, and email is {}.".format(name, record["Phone"],record["Email"]))    junaidRecord = phonebook.pop("Junaid Salman")  print("\nThis record has been popped out of the list:\n{}\n".format(junaidRecord))  for name, record in phonebook.items():  print("{}'s phone number is {}, and email is {}.".format(name, record["Phone"],record["Email"]))  #Deleting an already deleted record produces an error when prompted to be removed again.  **Output:**   |  | | --- | | Afaan Kamran's phone number is 0304-4657625, and email is fa20-bce-053@cuilahore.edu.pk.  Junaid Salman's phone number is 0301-5369018, and email is fa20-bce-044@cuilahore.edu.pk.  Rana Fahad Aman's phone number is 0349-4945272, and email is fa20-bce-021@cuilahore.edu.pk.  Junaid Salman's phone number is 0301-5369018, and email is fa20-bce-044@cuilahore.edu.pk.  Rana Fahad Aman's phone number is 0349-4945272, and email is fa20-bce-021@cuilahore.edu.pk.  This record has been popped out of the list:  {'Phone': '0301-5369018', 'Email': 'fa20-bce-044@cuilahore.edu.pk'}  Rana Fahad Aman's phone number is 0349-4945272, and email is fa20-bce-021@cuilahore.edu.pk. | |

**In-Lab Task 4**

**{Arithmetic Operators }**

|  |  |
| --- | --- |
| **Code:**  number = 1+2\*3/4.0  print(number)  remainder = 11%3  print(remainder)  squared = 7\*\*(2)  print(squared)  cubed = 2\*\*(3)  print(cubed)  **Output:**   |  | | --- | | 2.5  2  49  8 | |

**{List Operators }**

|  |  |
| --- | --- |
| **Code:**  evenNumbers = [2,4,6,8]  oddNumbers = [1,3,5,7]  Numbers = oddNumbers+evenNumbers  print(Numbers)  print(evenNumbers\*3)  x = object()  y = object()  xList = [x]  yList = [y]  concatList=[]  print(f"xList contains {len(xList)} objects.")  print(f"yList contains {len(yList)} objects.")  print(f"Joint List contains {len(xList)+len(yList)} objects.")  if (xList.count(x)==10 and yList.count(y)==10):  print("Almost there...")  if (concatList.count(x)==10 and concatList.count(y)==10):  print("Great!")  **Output:**   |  | | --- | | [1, 3, 5, 7, 2, 4, 6, 8]  [2, 4, 6, 8, 2, 4, 6, 8, 2, 4, 6, 8]  xList contains 1 objects.  yList contains 1 objects.  Joint List contains 2 objects. | |

**In-Lab Task 5**

**{String Operators }**

|  |  |
| --- | --- |
| **Code:**  mylist = []  print("Hello, World!")  stringToPrint = "Hello "  print(stringToPrint\*7)  **Output:**   |  | | --- | | Hello, World!  Hello Hello Hello Hello Hello Hello Hello | |

**In-Lab Task 6**

|  |  |
| --- | --- |
| **Code:**  x=2  print(x==2)  print(x==3)  print(x<3)  name = "Fahad"  print(name=="Fahad" and x==2)  print(name=="Fahad" or name=="John")  print(name in ["Fahad","John","Jess"])  if x>2:  print("Testing 'x'")  print("x > 2")  if x==2:  print("x==2")  **Output:**   |  | | --- | | True  False  True  True  True  True  x==2 | |

**In-Lab Task 7**

|  |  |
| --- | --- |
| **Code:**  x = 2  y = 10  if (x > 2):  print("x > 2")  elif(x == 2 and y > 50):  print("x == 2 and y > 50")  elif(x < 10 or y > 50):  print("x < 10 or y > 50")  else:  print("Nothing Worked!")    nameList1 = ["John","Jill"]  nameList2 = ["John","Jill"]  print(not(nameList1==nameList2))  name2 = "John"  print(nameList1 == nameList2)  print(nameList1 is nameList2)  **Output:**   |  | | --- | | x < 10 or y > 50  False  True  False | |

**In-Lab Task 8**

|  |  |
| --- | --- |
| **Code:**  numericData = []  for number in numericData:  result = number\*2  print(result)  text = "Hello, World!"  for char in text:  print(char)  newText = ""  for char in text:  if char.isalpha():  newText+=char.upper()  else:  newText +=char  print(newText)  numericData = []  for i in range(1,11):  numericData.append(i)  print(numericData)  **Output:**   |  | | --- | | H  e  l  l  o  ,    W  o  r  l  d  !  HELLO, WORLD!  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] | |

**In-Lab Task 9**

|  |  |
| --- | --- |
| **Code:**  #For Numeric Data  count = 1  while count<=5:  print(count)  count += 1  print("")  #For Strings  text = "Hello"  index = 0  while index < len(text):  print(text[index])  index += 1  print("")  #For Dictionaries  studentGrades = {"Alice":92, "Bob":85, "Charlie":78}  keys = list(studentGrades.keys())  index = 0  while (index<len(keys)):  key = keys[index]  value = studentGrades[key]  print(f"{key}: {value}")  index+=1  print("")  **Output:**   |  | | --- | | 1  2  3  4  5  H  e  l  l  o  Alice: 92  Bob: 85  Charlie: 78 | |

**Post-Lab**

**Post-Lab Task**

|  |  |
| --- | --- |
| **Code:**  # Initialize empty lists to store student names and their corresponding grades  studentNames = []  studentGrades = []  # Input grades for at least seven students  while len(studentNames) < 7:  name = input("Enter student name: ")  if not name.replace(" ", "").isalnum():  print("Invalid name input. Please enter a name with letters and spaces only.")  continue # Skip this iteration and ask for the name input again  try:  grade = float(input("Enter student's grade: "))  except ValueError:  print("Invalid grade input. Please enter a valid number.")  continue # Skip this iteration and ask for grade input again  studentNames.append(name)  studentGrades.append(grade)  # Function to calculate the average grade  def calculateAverageGrade(grades):  total = sum(grades)  return round((total / len(grades)), 3)  # Function to categorize grades  def categorizeGrade(grade):  if grade >= 90:  return "Excellent"  elif grade >= 80:  return "Very Good"  elif grade >= 70:  return "Good"  else:  return "Needs Improvement"  # Calculating the average and displaying  averageGrade = calculateAverageGrade(studentGrades)  print("\nAverage Grade:", averageGrade)  # Display the data of all students  print("\nList of Students and their respective Grades:")  for i in range(len(studentNames)):  name = studentNames[i]  grade = studentGrades[i]  print(f"{name}: {grade} ({categorizeGrade(grade)})")  # Search student records  while True:  searchName = input("\nEnter a student name to search for their grade (or 'quit' to exit): ")  if searchName == 'quit':  break  if searchName in studentNames:  index = studentNames.index(searchName)  grade = studentGrades[index]  print(f"{searchName}'s Grade: {grade} ({categorizeGrade(grade)})")  else:  print(f"{searchName} not found in the list of students.")  **Output:**   |  | | --- | | Enter student name: Rana Fahad Aman  Enter student's grade: 92  Enter student name: Daud Hassan  Enter student's grade: 84  Enter student name: Afaan Kamran  Enter student's grade: 67  Enter student name: Malaika Asghar  Enter student's grade: 55  Enter student name: Junaid Salman  Enter student's grade: 33  Enter student name: Aashir  Enter student's grade: 90  Enter student name: Ahsan Yousaf  Enter student's grade: 77  Average Grade: 71.143  List of Students and their respective Grades:  Rana Fahad Aman: 92.0 (Excellent)  Daud Hassan: 84.0 (Very Good)  Afaan Kamran: 67.0 (Needs Improvement)  Malaika Asghar: 55.0 (Needs Improvement)  Junaid Salman: 33.0 (Needs Improvement)  Aashir: 90.0 (Excellent)  Ahsan Yousaf: 77.0 (Good)  Enter a student name to search for their grade (or 'quit' to exit): Rana Fahad Aman  Rana Fahad Aman's Grade: 92.0 (Excellent)  Enter a student name to search for their grade (or 'quit' to exit): Daud Hassan  Daud Hassan's Grade: 84.0 (Very Good)  Enter a student name to search for their grade (or 'quit' to exit): Afaan Kamran  Afaan Kamran's Grade: 67.0 (Needs Improvement)  Enter a student name to search for their grade (or 'quit' to exit): quit | |