**BHARTIYA VIDYA BHAVAN’S**

**SARDAR PATEL INSTITUTE OF TECHNOLOGY**

**Bhavan’s Campus, Munshi Nagar, Andheri (West), Mumbai – 400058-India**

**ISE EVALUATION SHEET**

**CLASS: F.Y. SEM: II BRANCH: IT ACADEMIC YEAR:2019-20**

**COURSE: Python Programming Lab COURSE CODE: ESL25**

**STUDENT NAME: Mohsin Chougale ROLL NO: 2019140013**

|  |  |
| --- | --- |
| **Experiment No:** | **7** |
| **Aim:** | **To use the concepts of Classes and Objects for modelling applications in Python** |
| **Problem Statement** | Create a class called Student. Each student must have a first name, last name, uid, class year. Write a method which compares two student objects and prints them in the ascending order of names. Create any 2 student objects and test the above method. |
| **Program** | class Student():  def \_\_init\_\_(self,first, last, uid, year):  self.first = first  self.last = last  self.uid = uid  self.year = year    def compare():    for i in range(0,len(student1.first)):  if( student1.first[i] < student2.first[i] ):  print(student1.first , student1.last ,student1.uid , student1.year)  print()    print(student2.first , student2.last ,student2.uid , student2.year)  break    elif( student1.first[i] == student2.first[i] ):  continue    else:    print(student2.first , student2.last ,student2.uid , student2.year)  print()  print(student1.first , student1.last ,student1.uid , student1.year)  break  student1 = Student("Yash","Jain",12,2020)  student2 = Student("Ravi","Shankar",15,2020)  print("The details of students printed according to conditions are as follows :-")  print()  compare() |
| **Output** | **The details of students printed according to conditions are as follows :-**  **Ravi Shankar 15 2020**  **Yash Jain 12 2020** |

|  |  |
| --- | --- |
| **Problem Statement** | Create a class called Date() with the following specification:  I. Class variables: year\_min(1000),year\_max(9999),str\_days(list of days as str ―Sunday‖, Monday‖, ―Tuesday‖ etc.), str\_months(list of months as str— [―January‖, ―February‖,...]), days\_in\_months(list of size 12 with count of each month’s days---[31,28,31,30,31.....])  II. Object Attributes: day(1-31), month(1-12), year(1000-9999)  III. Methods:   1. setDate() #sets the entire Date object including yr, month and day 2. setDay() #sets the entire Date object including yr, month and day 3. setMonth() #sets the entire Date object including yr, month and day 4. setYear() #sets the entire Date object including yr, month and day 5. getYear() #returns the year of a Date object 6. getMonth() #returns the month of a Date object 7. getDay() #returns the day of a Date object 8. nextDay() # returns the day following the current Date’s day 9. previousDay()# returns the day before the current Date’s day 10. nextMonth()# returns the month following the current Date’s month 11. previousMonth() #returns previous month from the current Date 12. nextYear() #returns the next year from the current Date 13. previousYear() # returns the previous year from the current Date 14. printDateWords() # print current Date in a specific format   Ex. If Date is (16-03-2020),  If format=1◊ Monday, March 16, 2020  If format=2◊ 16-March-2020  (Assume several Date formats and implement the method) |
| **Program** | class Date():  year\_min = 1000  year\_max = 9999  str\_days = ["Sunday", "Monday", "Tuesday","Wednesday","Thursday","Friday","Saturday"]    str\_months = ["January","February","March","April","May","June","July","August","September","October","November","December"]  days\_in\_months = [31,28,31,30,31,30,31,31,30,31,30,31]        def \_\_init\_\_(self,day,month,year):  self.day = day  self.month = month  self.year = year      def setDate(self,day,month,year):  self.day = day  self.month = month  self.year = year    def setDay(self,day):  self.day = day    def setMonth(self,month):  self.month = month    def setYear(self,year):  self.year = year    def getDay(self):  return self.day    def getMonth(self):  return self.month    def getYear(self):  return self.year    def nextDay(self):  return self.day + 1    def previousDay(self):  return self.day - 1    def nextMonth(self):  return self.month + 1    def previousMonth(self):  return self.month - 1    def nextYear(self):  return self.year + 1    def previousYear(self):  return self.year - 1    def printDatewords(self,format1):  self.format1 = format1  if self.month == 1 :  x = 11  elif self.month == 2:  x = 12  else:  x = self.month -2    #For Jan and Feb  if self.month == 1 or self.month == 2 :    w = ( self.day + int(2.6\*x - 0.2) - 2 \*( (self.year-1) //100) + (self.year-1) % 100 + int( ( (self.year-1) % 100) /4 )  +( (self.year-1) //400) ) % 7    # (k +[2.6\*m-0.2] -2C +Y +[Y/4] + [C/4])mod7  # For other months  else:  w = ( self.day + int(2.6\*x - 0.2) - 2 \*(self.year//100) +self.year%100 + int( (self.year % 100) /4 )  +(self.year//400) ) % 7      D = self.str\_days[w]  M = self.str\_months[(self.month-1)]  Y = str(self.year)  DA = str(self.day)    #Printing Date in different formats  if format1 == 1:    print(D,",",DA,"th",M,"",Y)    elif format1 == 2:  print(D,",",Y," ",DA,"th",M)    else:  print(D,",",M,"",DA,"nd",Y)        print("Enter the following aspects of a date")  a = int(input("Day "))  b = int(input("Month "))  c = int(input("Year "))  f = int(input("Enter the format "))  obj = Date(a,b,c)  obj.setDate(a,b,c)  obj.setDay(a)  obj.setMonth(b)  obj.setYear(c)  print("The current day is")  print( obj.getDay() )  print()  print("The current month is ")  print( obj.getMonth() )  print()  print("The current year is")  print( obj.getYear() )  print()  print("The next day is")  print( obj.nextDay() )  print()  print("The previous day was")  print( obj.previousDay() )  print()  print("The next month is")  print( obj.nextMonth() )  print()  print("The previous month was")  print( obj.previousMonth() )  print()  print("The next year is")  print( obj.nextYear() )  print()  print("The previous year was")  print( obj.previousYear() )  print()  print("The date in required format is as follows")  obj.printDatewords(f) |
| **Output** | **Enter the following aspects of a date**  **Day 23**  **Month 07**  **Year 2001**  **Enter the format 1**  **The current day is**  **23**    **The current month is**  **7**    **The current year is**  **2001**    **The next day is**  **24**    **The previous day was**  **22**    **The next month is**  **8**    **The previous month was**  **6**    **The next year is**  **2002**    **The previous year was**  **2000**    **The date in required format is as follows**  **Monday , 23 th July  2001**  **##Attempt 2**  **Enter the following aspects of a date**  **Day 16**  **Month 3**  **Year 2020**  **Enter the format 2**  **The current day is**  **16**    **The current month is**  **3**    **The current year is**  **2020**    **The next day is**  **17**    **The previous day was**  **15**    **The next month is**  **4**    **The previous month was**  **2**    **The next year is**  **2021**    **The previous year was**  **2019**    **The date in required format is as follows**  **Monday , 2020   16 th March** |

|  |  |
| --- | --- |
| **Problem Statement** | Demonstrate any real-world example which involves inheritance using the concept of Python inheritance in terms of parent-child relationship. (You may use any type of inheritance-single, multiple, multi-level) |
| **Program** | class Parent():  def \_\_init\_\_(self,father,mother):  self.father = father  self.mother = mother        class Name1(Parent):  def \_\_init\_\_(self,father,mother,name1):  self.\_\_name1 = name1  Parent.\_\_init\_\_(self,father,mother)    def getName1(self):  print("My father's name is ",self.father)  print("My mother's name is ",self.mother)  print("My name is",self.\_\_name1)  print("\n")  class Name2(Parent):  def \_\_init\_\_(self,father,mother,name2):  self.\_\_name2 = name2  Parent.\_\_init\_\_(self,father,mother)    def getName2(self):  print("My father's name is ",self.father)  print("My mother's name is ",self.mother)  print("My name is",self.\_\_name2)    a = input("Enter the father's name ")  b = input("Enter the mother's name ")  c = input("Enter the name of 1st sibling ")  d = input("Enter the name of 2nd sibling ")  print("\n")  o1 = Name1(a,b,c)  o2 = Name2(a,b,d )  o1.getName1()  o2.getName2() |
| **Output** | **Enter the father's name  Jason**  **Enter the mother's name  Mary**  **Enter the name of 1st sibling  Jack**  **Enter the name of 2nd sibling  Angela**      **My father's name is  Jason**  **My mother's name is  Mary**  **My name is Jack**      **My father's name is  Jason**  **My mother's name is  Mary**  **My name is Angela** |
| **Conclusion** | **We learnt how to use the concepts of Classes and Objects for modelling applications in Python** |