Examination Roll No. → 20234757062

Name → Anjali Singh

Date & Time of Examination → 27/03/21
9:30 - 12:30

Unique Paper Code → 223401101

Name of Programme → MCA (CBCS PG)

Semester → Ist

Title of the Paper → Object Oriented Programming

Email id → 200105@cs.du.ac.in

Mobile Number → 8770061023

Total Pages used in Question → 5

```
class Date:
    count = 0   # class attribute count
    def __init__(self, day, month, year):
# constructor for class ~~Employee~~ Date
        if day < 0 or day > 31 :
            raise Exception("Days should be
                                in between 1-31")
#if day or months are invalid then raise
exception
        if month < 0 or month > 12 :
            raise Exception("Months b/w 1-12")

        Date.count += 1     # constructor for
        self._day = day     class date whose
        self._month = month attributes are
        self._year = year   day, month, year

    def __str__(self) : # string fun overloading
        return f"{self._day}/{self._month}/
                                {self._year}"

    def __del__(self) : # overload del operator
        Date.count -= 1

    def __le__(self, d2) : # overloading
        if self._year > d2.y : # to check if
                                date is valid or not
            return False
```

```
if self.__month > d2.m and self.__year == d2.y:
    return False

if self.__day > d2.d and self.__month == d2.m:
    return False

return True # else return True i.e. date is valid
@classmethod # property decorator
def objectcount(cls):
    return cls.count

@property # getter for Date day
def day(self):
    return self.__day

@property # getter for year
def year(self):
    return self.__year

@property # getter for date month
def month(self):
    return self.__month

Class Employee: # class employee
    count = 0
    def __init__(self, name, hdate, sal):
# constructor for class employee
        Employee.__count += 1
        self.__name = name
```

```
        self.__hdate = hdate
        self.__sal = sal

    def __str__(self):
        return f'''                # return string of user
            Employee details :          information
            Name : {self.__name}
            Date of hiring : {self.__hdate}
            Salary : {self.__sal}
            '''

    def __del__(self):  # overloading del operator
        Employee.count -= 1

    @classmethod
    def object count(cls):
        return cls.count

    @Property
    def name(self):  # getter for object name
        return self.__name

    @property
    def date(self):  # getter for object date
        return self.__hdate

    @property
    def salary(self):  # getter for object sa
        return self.__sal
```

```
def exp(empobj, dateobj):
# to calculate the experience of employee object
    hdate = empobj.hdate

    if date1 <= hdate:
        raise Exception("Invalid hdate")

    if hdate.day <= dateobj.day:
        expday = dateobj.day - hdate.day
        expmonth = dateobj.month - hdate.month

    else:
        expday = dateobj.day - hdate.day + 30
        expmonth = dateobj.month - hdate.month - 1

    expyear = dateobj.year - hdate.year
# to calculate exp year
    return Date(expday, expmonth, expyear)
# returning Date i.e. exp of employee object

employee_list = []  # to store employee objects
print("""Menu/n                # Menu driven
        1. Add employee
        2. employees experience
        3. total employees
        4. Exit
    """)
choice = int(input()).strip()
while (choice != 4):
    if choice == 1:                # user choice
        name = input('Name:')
```

```
        day, month, year, salary = map(int, input("
            Enter space separated ").split())

        try: # try except to take invalid details
            employee_list.append(Employee(name,
                Date(day, month, year), salary))

        except Exception:
            print("Invalid Input")

        elif choice == 2:
            day, month, year = map(int, input().split())
# experience of
emp obj in current Date = Date(day, month, year)
format given for exp in employee list:
            print(exp)

        elif choice == 3:
            print(f"total no. of employees : {employee
# count of Employee class                    .count}")
        elif choice == 4:
            break # to break the loop
        else:
            print('Enter choice')
            choice = int(input(choice).strip())
```