Unique Paper Code	:	**32345104**
Name of the Paper	:	**Programming using Python**
Name of the Course	:	**Computer Science: Generic Elective for Honours**
Semester	:	**I**
Year of Admission	:	**2019 onwards**

Duration: **3 Hours** Maximum Marks: **75**

Attempt any **four** questions.
All questions carry equal marks.

Question 1.

- Write Python functions for the following:
  - Return 1 if a given number is prime or 0 otherwise.
  - Return the number of words in a given sentence.

- Which of the following are valid identifiers in Python? Justify your answer.
  - First Number
  - List#Elements
  - 54Number
  - _FourthNumber
  - Pass
  - del
- Given a file `StudentData.txt` containing student name, course and marks of students. Write a Python function that takes a parameter `n`, reads the data of first `n` students from the file `StudentData.txt` and copies it to another file `Replica.txt`.

- What will be the output generated by the following Python statements? Justify your answer.

```
list1 = list({'GE1': 'Python', 'GE2': 'DBMS',
              'GE3':'CN', 'GE4': 'ISCL'})
print(list1)
print(type(list1))
```

Question 2. What will be the output of the following Python code segments? Justify your answers.

- ```
  marks = 67
  def func(marks, IA):
      IA = 23
      marks = marks + IA
      return marks

  print(func(57))
  print(func(53, 24))
  print(func(67, 12))
  print(func(67, 20, 5))
  ```

- ```
  func = lambda x : x * y
  print(func(5,3))
  ```

- ```
  func = lambda x : x ** x
  print(func(2))
  ```

- ```
  func = lambda x : x * x, y : x + 10
  print(func(4))
  ```

- ```
  a = 0
  while a < 20:
      a += 4
      print(a, end=",")
  else:
      print("In else block")
  ```

- ```
  list1 = [j for i in range(1,6,2) for j in
  range(1, i+2, 2)]
  print(list1)
  ```

Question 3.

- Apply selection sort to the list below. Show the modified list and the index position of minimum index after each iteration of the algorithm:

```
['Siya', 'Anjali', 'Ritu', 'Zoya', 'Rita', 'Payal']
```

What will be the output of the following? Justify your answers.

- ```
  list2 = ['Eleven', 'Twelve', 'Fourteen',
  'Fifteen']
  print(list2.sort())
  list2.sort()
  print(list2)
  ```

- ```
  set1 = {'Python', 'Java', 'R', 'Ruby', 'Pearl'}
  set1.add('PHP')
  set1.add('Ruby')
  print(set1)
  ```

- ```
  tuple1 = (['a', 'b', 'c', 'd'], [2,3,4], ['x',
  'y', 'z'], 9)
  tuple1.append(12)
  print(tuple1)
  ```

- ```
  dict1 = {'Subjects': ['Physics', 'Chemistry',
  'Maths', 'CS'], 'Marks': [93, 97, 95, 92]}
  print(dict1.get('StudentMarks', None))
  print('Music' in dict1)
  print(dict1.keys())
  ```

Question 4. Write Python functions with appropriate comments for the following, taking an integer `n` as an argument:

- To generate sum of the following series for `n` terms:
  1 + 2/2! + 3/3! + …….+ n/n!
  Use appropriate assertions where needed.

- To print all the Armstrong numbers less than `n`. (An Armstrong number is a number whose sum of the cubes of the digits is equal to the number itself. For example, $370 = 3^3 + 7^3 + 0^3$).

- To find the smallest number `n` such that `n*n` > 12000.

- To print the following pattern, here `n` is the number of rows in the pattern. For example, for `n` = 4 it prints:

```
$ $ $ $
  $ $ $
    $ $
      $
```

Question 5.

- What will be the output generated by the following statements? Justify your answers.

```
o 5//2 and 'Hello' > 'Hi'
o 'welcome' * 3 + 7
o y **= 2
o 15 % 5 + 9 - 42 * 2 / 3
o 20 & 35
o 'Ramesh' > 'Mukesh' or 'harry' < 'Harry'
```

- Show the contents of the stack after every operation during evaluation of the postfix expression: `526*93/-*`

- What will be the output generated by these statements? Justify your answer.

```
list3 = ['Abhay', 'Sarita', 'Meenu',
'Shamishta']
list3.insert('Meenakshi')
print(list3)
list3.insert(len(list3)-1, 'Meenakshi')
print(list3)
```

- Identify and describe three different types of errors that may be raised while executing the following code:

```
Percentage = (marks/ total) * 100
```

Question 6. Define a class `Flat_Maintenance` that keeps a record of the payments made by the members of the society for flat maintenance. The class should contain the following:

- data members for `Flat_Maintenance` include: `owner_name`, `flat_number`, `tower_name`, `maintenance_amount`, `month`, and `year`.
- A data member `count` keeps track of number of objects created for this class. Display the value of `count` every time an object is created when a payment is made by a member.

- Define following member functions for the class:
  - a `constructor` function to initialize the members.
  - __str__ function to display the complete details of an object of `Flat_Maintenance`, along with the number of objects of the class.
- Write statements for the following:
  - Take appropriate input values from the user to create an object `Owner1` of this class. (For example: Ramesh of flat number 006 in tower C and wants to pay 3500 for the month of November 2021).
  - Display the details of the object `Owner1`.