

INTRODUCTION TO COMPUTER PROGRAMMING WITH PYTHON

SEMESTER ONE TEST

Duration: 2 hours

Instructions:

- 1. This test has to be submitted from GitHub. An invitation link will be shared with you to get started as soon as you can.**
- 2. All the answers should be included in the editor, I have selected VScode as the online editor for use in Github.**
- 3. Use comments to describe your solutions**
- 4. Each question should have a number as a comment.**
- 5. Create a different python file for each question e.g question-1.py, question-2.py etc**
- 6. You may choose to follow the approach given or you may decide to use your own approach.**

1. Strings: Reversing a string

Given a string, write a python program to reverse words of a given string.

For example:

Input : str = geeks quiz practice code

Output : str = code practice quiz geeks

2. Dictionaries: Program to create grade calculator in Python

Given different scoring marks of students.

1. Jack's dictionary

```
jack = { "name": "Jack Frost",  
        "assignment" : [80, 50, 40, 20],  
        "test" : [75, 75],  
        "lab" : [78.20, 77.20]  
    }
```

2. James's dictionary

```
james = { "name": "James Potter",  
          "assignment" : [82, 56, 44, 30],  
          "test" : [80, 80],  
          "lab" : [67.90, 78.72]  
    }
```

3. Dylan's dictionary

```
dylan = { "name" : "Dylan Rhodes",  
          "assignment" : [77, 82, 23, 39],  
          "test" : [78, 77],  
          "lab" : [80, 80]  
    }
```

4. Jessica's dictionary

```
jess = { "name" : "Jessica Stone",  
         "assignment" : [67, 55, 77, 21],  
         "test" : [40, 50],  
         "lab" : [69, 44.56]  
    }
```

5. Tom's dictionary

```
tom = { "name" : "Tom Hanks",  
        "assignment" : [29, 89, 60, 56],  
        "test" : [65, 56],  
        "lab" : [50, 40.6]
```

}

Find grades for each student. The test score is an average of the respective marks scored in assignments, tests and lab-works. The final test score is assigned using the below formula.

10 % of marks scored from submission of Assignments

70 % of marks scored from Test

20 % of marks scored in Lab-Works

Grade will be calculated according to :

1. score ≥ 90 : "A"

2. score ≥ 80 : "B"

3. score ≥ 70 : "C"

4. score ≥ 60 : "D"

Also, calculate the total class average and letter grade of class.

3. Object Oriented Programming: Student management system in Python

Write a program to build a simple Student Management System using Python which can perform following operations:

1. Accept
2. Display
3. Search
4. Delete
5. Update

Approach:

1. **Accept** – This method takes details from the user like name, roll number, and marks for two different subjects.

```
# Method to enter new student details
def accept(self, Name, Rollno, marks1, marks2 ):
    # Creates a new class constructor
    # and pass the details
    ob = Student(Name, Rollno, marks1, marks2 )

    # list containing objects of student class
    ls.append(ob)
```

2. **Display** – This method displays the details of every student.

```
# Function to display student details
def display(self, ob):
    print("Name  :", ob.name)
    print("RollNo :", ob.rollno)
    print("Marks1 :", ob.m1)
    print("Marks2 :", ob.m2)
```

```
print("\n")
```

3. Search – This method searches for a particular student from the list of students.

This method will ask the user for roll number and then search according to the roll number

```
# Search Function
def search(self, rn):
    for i in range(ls.__len__()):
        # iterate through the list containing
        # student object and checks through
        # roll no of each object
        if(ls[i].rollno == rn):
            # returns the object with matching
            # roll number
            return i
```

4. Delete – This method deletes the record of a particular student with a matching roll number.

```
# Delete Function
def delete(self, rn):
    # Calls the search function
    # created above
    i = obj.search(rn)
    del ls[i]
```

5. Update – This method updates the roll number of the student.

This method will ask for the old roll number and new roll number. It will replace the old roll number with a new roll number.

```
# Update Function
def update(self, rn, No):
    # calling the search function
    # of student class
    i = obj.search(rn)
    ls[i].rollno = No
```

4. File Handling: Reading data from one file to another

Create a file called **read.txt**. Copy the text below and save it in that file. Create an empty file called **write.txt**. Write a python program to read the contents of **read.txt** file and write it into **write.txt**.

This course introduces the core python programming basics—including data types, control flow structures, data structures, and functions. The course also discusses the fundamental principles of Object-Oriented Programming. Students will solve problems, explore real-world software development challenges, and create practical and contemporary applications. The course will also introduce students to version control using Git and GitHub. The students will use this throughout the course to submit their assignments, and for collaboration and code management.