



Columbia College
Vancouver, Canada

Introduction to Computer Science and Programming 1
CSCI120

Chapter12: Object-Oriented Programming

Assignment 12-1

Note: This document has been designed and developed as part of an initiative for creating an OER (Open Education Resource) package for the course CSCI 120 at Columbia College.

Please contact Alireza.davoodi@gmail.com for any comment, modification, and questions.

Terms of use: Please feel free to customize this document as needed

Last Modified: July 2022



If it is a group assignment, please add the information here

# of Students in the Group:		
Student 1	<i>First name, last name</i>	<i>Student-ID</i>
Student 2	<i>First name, last name</i>	<i>Student-ID</i>
Student 3	<i>First name, last name</i>	<i>Student-ID</i>
Student 4	<i>First name, last name</i>	<i>Student-ID</i>

Requirements

- Please use meaningful name for your classes, properties and methods
- Try to reuse your solutions as much as possible.
- You can define the classes in one file or separate files.
- Define one main function (one for each file)
- Define some instances (if needed from the classes you have defined) and try them in the main function.
- Do not use methods, functions, statements that we have not covered in the previous lectures.

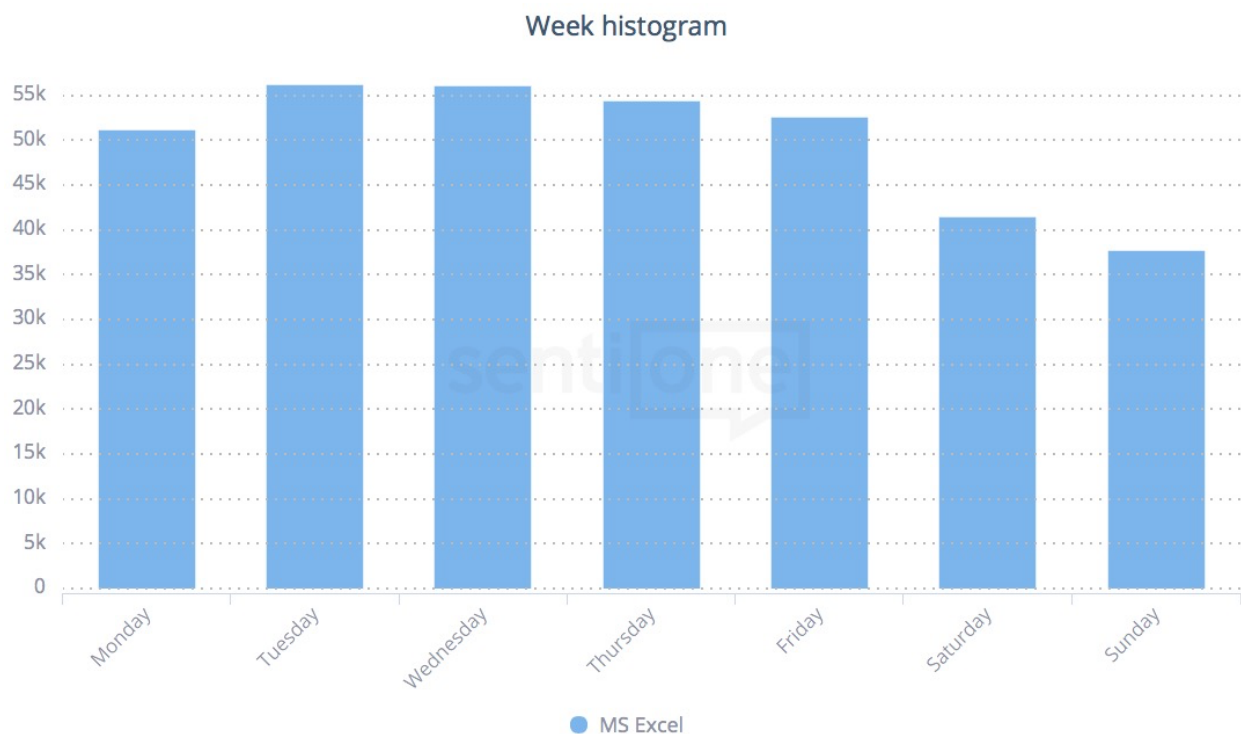


Problem1:

- Define a class and a static method that receives a string (word) and reverses it and returns it.

Problem2:

- Design a class that represents a histogram like the one in the image below. The X axis represents the days of the week and then the Y axis represents the sales.
- Add an instance method which receives a day and return the corresponding Y value.



Problem3:

- Define and design a class with a static method called *convert*, which receives a positive number and a base number and converts the number to the given base and returns it as a string.
 - o For instance: convert (9, 2) returns 1001



Problem4:

- Look around your classroom or your bedroom and come up with 3 classes that you see there.
- For each class define 3 instance Variables
- For each class define 1 static variable
- For each class define an accessor and a mutator for each instance variable
- For each class define 2 instance methods in addition to the accessors and the mutators.
- For each class define a constructor which initializes all its instance variables.

Problem5:

- Write a main function and inside the function write a test scenario for each of the above classes. Your test scenario should include creating one or two instances of each class, call some of its accessors and mutators and its instance variables.
- Example:

```
class Student:

    _studentIDCounter = 1000

    def __init__(self, firstName, lastName, address, age):
        self._listOfCourse = []
        self._firstName = firstName
        self._lastName = lastName
        self._address = address
        self._age = age
        Student._studentIDCounter = Student._studentIDCounter + 1
        self._studentID = Student._studentIDCounter

    def _getFirstName(self):
        return self._firstName

    def _getLastName(self):
        return self._lastName

    def _getAddress(self):
        return self._address
```



```
def _getAge(self):
    return self._age

def addCourseGrades(self, grade):
    self._listOfCourse.append(grade)

def calculateAverage(self):
    if len(self._listOfCourse) > 0:
        sum = 0
        for item in self._listOfCourse:
            sum = sum + item
        average = sum / len(self._listOfCourse)
        return average
    else:
        return

def getStudentID(self):
    return self._studentID

def printStudentProfile(self):
    print("=====")
    print(self._firstName)
    print(self._lastName)
    print(self._age)
    print(self._address)
    print(self.getStudentID())
    print(self.calculateAverage())
```

class TesStudent:

```
def test(self):
    student1 = Student("Peter", "Mak", "Vancouver", "29")
    student2 = Student("David", "Cameron", "Burnaby", "30")
    student1.printStudentProfile()
    student2.printStudentProfile()

    student1.addCourseGrades(80)
    student1.addCourseGrades(73)
    student1.addCourseGrades(85)

    student2.addCourseGrades(48)
    student2.addCourseGrades(90)
```



```
student1.printStudentProfile()
student2.printStudentProfile()

def main():
    testStudent = TestStudent()
    testStudent.test()

main()
```

Problem6:

- Write a python program with the following description:
- Define a class called MyCustomList.
- The class has an instance variable called myList which is a list of integers.
- Define the following instance methods for the class:
 - o addItem: It has an input of type int and it add the number to the list if this number already does not exist in the list. If it exists it will ignore it.
 - o calculateSum: It has no input and will return the sum of all numbers in the list.
 - o calculateMax: it has no input and will return the maximum number of the list.
 - o printList: It has no input but print the current content of the list to the out.
 - o Remember to define a constructor for this class.
- Define another class called Test MyCustomList. This class has a static method called testMyCustomList. This method is used to contain a test scenario for the MyCustomList.
- In another python file, define a main function and use the TestClass to test the MyCustomList class.

Good Luck ☺