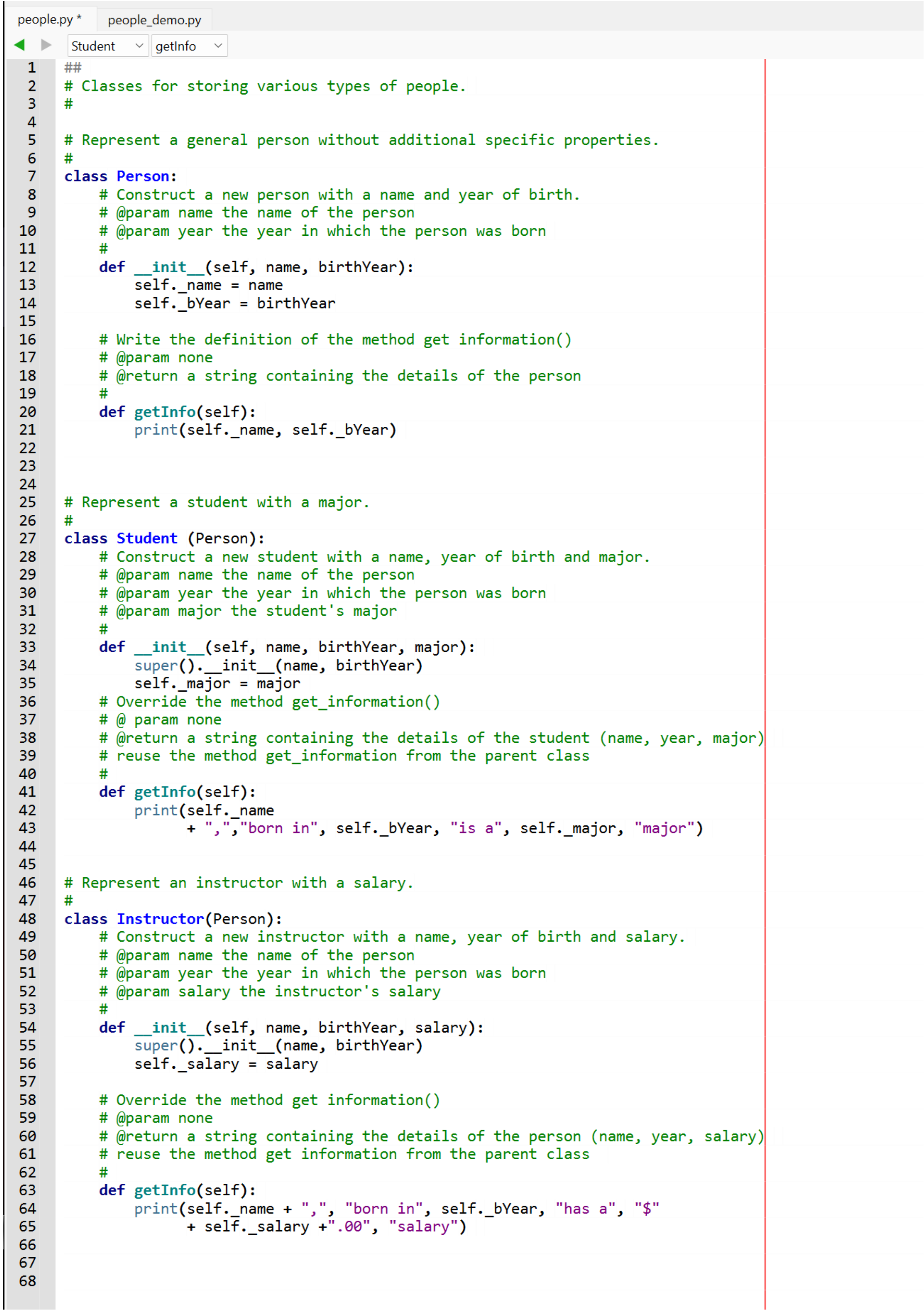
Thomas Nguyen, Albert Nguyen
CS131-47853 MW 06:00 PM – 08:30 PM
Lab 10

Lab 10

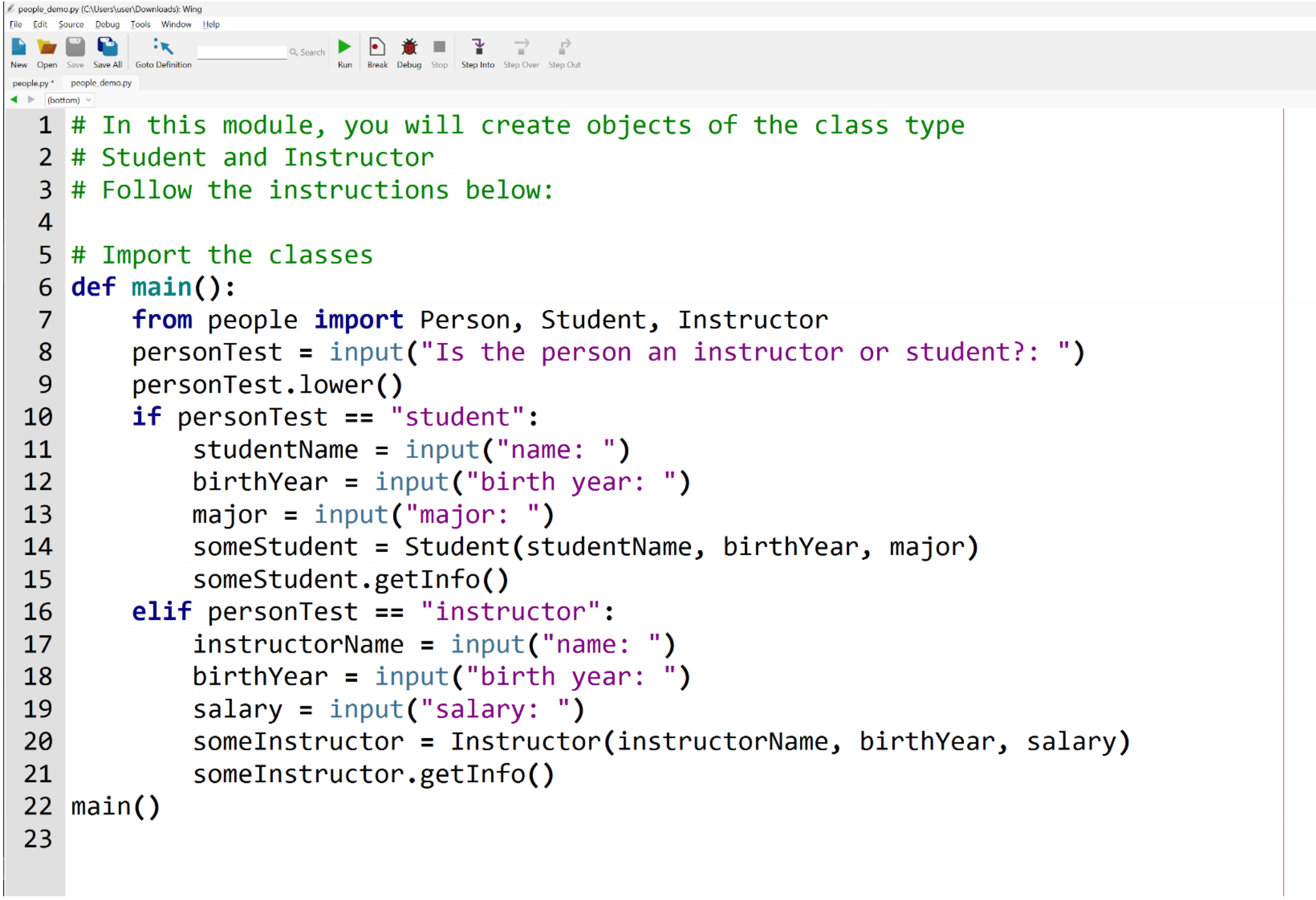
Lab 10 - Code

people.py



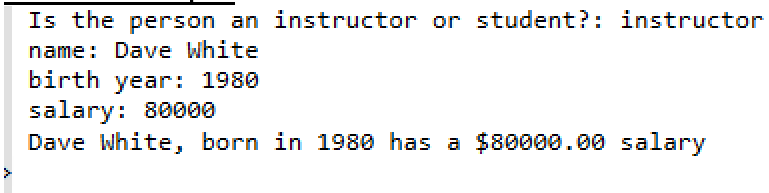
```python
##
# Classes for storing various types of people.
#

# Represent a general person without additional specific properties.
#
class Person:
    # Construct a new person with a name and year of birth.
    # @param name the name of the person
    # @param year the year in which the person was born
    #
    def __init__(self, name, birthYear):
        self._name = name
        self._bYear = birthYear

    # Write the definition of the method get information()
    # @param none
    # @return a string containing the details of the person
    #
    def getInfo(self):
        print(self._name, self._bYear)



# Represent a student with a major.
#
class Student (Person):
    # Construct a new student with a name, year of birth and major.
    # @param name the name of the person
    # @param year the year in which the person was born
    # @param major the student's major
    #
    def __init__(self, name, birthYear, major):
        super().__init__(name, birthYear)
        self._major = major
    # Override the method get_information()
    # @ param none
    # @return a string containing the details of the student (name, year, major)
    # reuse the method get_information from the parent class
    #
    def getInfo(self):
        print(self._name
              + ",","born in", self._bYear, "is a", self._major, "major")


# Represent an instructor with a salary.
#
class Instructor(Person):
    # Construct a new instructor with a name, year of birth and salary.
    # @param name the name of the person
    # @param year the year in which the person was born
    # @param salary the instructor's salary
    #
    def __init__(self, name, birthYear, salary):
        super().__init__(name, birthYear)
        self._salary = salary

    # Override the method get information()
    # @param none
    # @return a string containing the details of the person (name, year, salary)
    # reuse the method get information from the parent class
    #
    def getInfo(self):
        print(self._name + ",", "born in", self._bYear, "has a", "$"
              + self._salary +".00", "salary")
```

people_demo.py

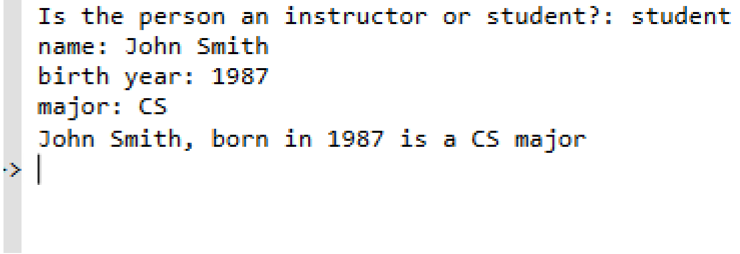

people_demo.py (C:\Users\user\Downloads): Wing
File Edit Source Debug Tools Window Help
New Open Save Save All Goto Definition Search Run Break Debug Stop Step Into Step Over Step Out
people.py * people_demo.py
(bottom)

```python
# In this module, you will create objects of the class type
# Student and Instructor
# Follow the instructions below:

# Import the classes
def main():
    from people import Person, Student, Instructor
    personTest = input("Is the person an instructor or student?: ")
    personTest.lower()
    if personTest == "student":
        studentName = input("name: ")
        birthYear = input("birth year: ")
        major = input("major: ")
        someStudent = Student(studentName, birthYear, major)
        someStudent.getInfo()
    elif personTest == "instructor":
        instructorName = input("name: ")
        birthYear = input("birth year: ")
        salary = input("salary: ")
        someInstructor = Instructor(instructorName, birthYear, salary)
        someInstructor.getInfo()
main()

```

Lab 10 - Output



```
Is the person an instructor or student?: instructor
name: Dave White
birth year: 1980
salary: 80000
Dave White, born in 1980 has a $80000.00 salary
>
```



```
Is the person an instructor or student?: student
name: John Smith
birth year: 1987
major: CS
John Smith, born in 1987 is a CS major
>
```

Lab 10 – Written Code

people.py

```
##
# Classes for storing various types of people.
#

# Represent a general person without additional specific properties.
#
class Person:
    # Construct a new person with a name and year of birth.
    # @param name the name of the person
    # @param year the year in which the person was born
    #
    def __init__(self, name, birthYear):
        self._name = name
        self._bYear = birthYear

    # Write the definition of the method get information()
    # @param none
    # @return a string containing the details of the person
    #
    def getInfo(self):
        print(self._name, self._bYear)

# Represent a student with a major.

#
class Student (Person):
    # Construct a new student with a name, year of birth and major.
    # @param name the name of the person
    # @param year the year in which the person was born
    # @param major the student's major
    #
    def __init__(self, name, birthYear, major):
        super().__init__(name, birthYear)
        self._major = major
    # Override the method get_information()
    # @ param none
    # @return a string containing the details of the student (name, year, major)
    # reuse the method get_information from the parent class
    #
    def getInfo(self):
        print(self._name + ",","born in", self._bYear, "is a", self._major, "major")
```

```
# Represent an instructor with a salary.
#
class Instructor(Person):
    # Construct a new instructor with a name, year of birth and salary.
    # @param name the name of the person
    # @param year the year in which the person was born
    # @param salary the instructor's salary
    #
    def __init__(self, name, birthYear, salary):
        super().__init__(name, birthYear)
        self._salary = salary

    # Override the method get information()
    # @param none
    # @return a string containing the details of the person (name, year, salary)
    # reuse the method get information from the parent class
    #
    def getInfo(self):
        print(self._name + ",", "born in", self._bYear, "has a", "$" + self._salary +".00", "salary")
```

Thomas Nguyen, Albert Nguyen
CS131-47853 MW 06:00 PM – 08:30 PM
Lab 10

people_demo.py

```python
# In this module, you will create objects of the class type
# Student and Instructor
# Follow the instructions below:

# Import the classes
def main():
    from people import Person, Student, Instructor
    personTest = input("Is the person an instructor or student?: ")
    personTest.lower()
    if personTest == "student":
        studentName = input("name: ")
        birthYear = input("birth year: ")
        major = input("major: ")
        someStudent = Student(studentName, birthYear, major)
        someStudent.getInfo()
    elif personTest == "instructor":
        instructorName = input("name: ")
        birthYear = input("birth year: ")
        salary = input("salary: ")
        someInstructor = Instructor(instructorName, birthYear, salary)
        someInstructor.getInfo()
main()
```