**HOMEWORK**

Write a complete implementation of OCCCPerson.

Write a program named TestPerson that performs the following tasks:

As always, provide source and executable on USB and source on paper.

* Display a friendly greeting to the user
* Prompt the user for the following information:
  + First name
  + Last name
  + Birth Year
  + Birth Month
  + Birth Day
* Create an instance of OCCCDate, d1, using the current time.
* Display that date using d1.toString()
* Create an instance of OCCCDate, d2, using the information provided.
* Display that date using d2.toString()
* Prompt the user for today’s date (month, day, year).
* Create an OCCCDate d3 using today’s date as provided by the user.
* Display the results of d1.equals(d3).
* Create a Person p1 using only the first and last name.
* Create a Person p2 using the first name, last name, and OCCCDate d2.
* Display the first name, last name, and age of that person as above.
* Display the results of p1.equals(p2).
* Create a Person p3 using the same name as p1 but date d3.
* Display the results of p1.equals(p3).
* Prompt the user for a new last name.
* Change the last name of p3 to that provided using setLastName().
* Display the state of p3 using the toString() function.
* Prompt the user for a student ID
* Create an OCCCPerson p4 using p3 and that student ID
* Create an OCCCPerson p5 using the first name, last name, d2, and the student ID

(but make the first and last name in ALL CAPS)

* Display the results of p4.equals(p5)
* Create an OCCCPerson p6 by simply stating that p6 = p5.
* Display the results of p5.equals(p6).
* Prompt the user for a new last name.
* Set the last name of p6 to that new last name using the setLastName() function.
* Display the state of both p5 and p6 using the toString() function.
* Create an OCCCPerson p7 using p6 (a copy constructor).
* Display the results of p6.equals(p7).
* Change the last name of p6.
* Display the state of p6 and p7 using the toString() function.