Work on this lab with a new partner: someone with whom you have not yet worked.

Write the code for the lab below and each partner must upload the zipped src folder to the learning hub (learn.bcit.ca 🡪 Activities 🡪 Assignments 🡪 Lab5) before the start of lesson 6.

Include your full name and your partner’s full name at the top of each file, using a Java Javadoc   
/\*\* @author \*/ tag (for example: /\*\* @author Paul Mills and Jason Wilder \*/).

Create a class called Person that has instance variables, setters (as appropriate), and getters for the following data members:

* First name
* Last name
* Birth year
* Married // let’s make it a String; “**yes**, **no**, or **divorced**”
* Weight in pounds
* Highest education level (can be only “high school”, “undergraduate”, or “graduate”)

Instance variables should be named as adjectives and nouns (e.g. married). Methods should be named as verbs (e.g. isMarried()).

Create several overloaded constructors, as follows:

1. One constructor takes all six values (above) as parameters
2. Another constructor takes five values as parameters, but sets the birth year automatically to the current year (use a public static final int CURRENT\_YEAR constant set to today’s year)
3. A third constructor takes three values as parameters, but sets the birth year automatically to the current year, the married status to “no”, and the highest education level to “high school”.

In the first two constructors, ensure that the highest education level is only one of the three specified, by calling a private method with the signature **private boolean isValidEducationLevel(String educationLevelToVerify)**, and also ensure that the married status is either “yes” or “no” or “divorced” with a similar private **isValidMarriageStatus** method. If the parameters are wrong, throw an IllegalArgumentException from the constructor. Note: to compare Strings you cannot use ==. Use the .equals() method to compare Strings:

if(married == “yes”) // **wrong**

if(married.equals(“yes”)) // right

Create overloaded methods for:

**printDetails()   
printDetails(boolean kilograms) // don’t use magic numbers!  
printDetails(boolean kilograms, boolean uppercase)**

All three methods print sentences in one of the following exact formats:

“Tiger Woods (divorced) was born in 1975, weighs 200.0 pounds, and has an undergraduate degree!”

Note: convert to kilograms for the second overloaded method if true is passed in.

Note: print the names out in UPPERCASE in the third overloaded method if true is passed in; otherwise use all lowercase:

“TIGER WOODS (divorced) was born in 1975, weighs 200.0 pounds, and has an undergraduate degree!” or

“tiger woods (divorced) was born in 1975, weighs 200.0 pounds, and has an undergraduate degree!” or

“TIGER WOODS (divorced) was born in 1975, weighs 90.9 kilograms, and has an undergraduate degree!” or

“tiger woods (divorced) was born in 1975, weighs 90.9 kilograms, and has an undergraduate degree!”

Other forms of the sentences would be:

“bill gates (married) was born in 1955, weighs 80.9 kilograms, and has an undergraduate degree!” or

“oprah winfrey (single) was born in 1954, weighs 82.9 kilograms, and has a high school diploma!”

**Note: Use String.format("%.1f", weightLb) to print to 1 decimal place.**

**Here is a sample run of my main method and its proper expected output:**

**public static void main(String[] args)**

**{**

**Person p1;**

**Person p2;**

**Person p3;**

**Person p4;**

**p1 = new Person("Tiger", "Woods", 1975, "divorced", 200, "undergraduate");**

**p1.printDetails();**

**p1.printDetails(true);**

**p1.printDetails(true, true);**

**p1.printDetails(true, false);**

**p1.printDetails(false, true);**

**p1.printDetails(false, false);**

**p2 = new Person("Jason", "Harrison", 2000, "no", 180, "graduate");**

**p2.printDetails();**

**p2.printDetails(true);**

**p2.printDetails(true, true);**

**p2.printDetails(true, false);**

**p2.printDetails(false, true);**

**p2.printDetails(false, false);**

**p3 = new Person("Santa", "Claus", 1000, "yes", 280, "high school");**

**p3.printDetails();**

**p3.printDetails(true);**

**p3.printDetails(true, true);**

**p3.printDetails(true, false);**

**p3.printDetails(false, true);**

**p3.printDetails(false, false);**

**}**

**Tiger Woods (divorced) was born in 1975, weighs 200.0 pounds, and has an undergraduate degree!**

**Tiger Woods (divorced) was born in 1975, weighs 90.9 kilograms, and has an undergraduate degree!**

**TIGER WOODS (divorced) was born in 1975, weighs 90.9 kilograms, and has an undergraduate degree!**

**tiger woods (divorced) was born in 1975, weighs 90.9 kilograms, and has an undergraduate degree!**

**TIGER WOODS (divorced) was born in 1975, weighs 200.0 pounds, and has an undergraduate degree!**

**tiger woods (divorced) was born in 1975, weighs 200.0 pounds, and has an undergraduate degree!**

**Jason Wilder (single) was born in 2000, weighs 180.0 pounds, and has a graduate degree!**

**Jason Wilder (single) was born in 2000, weighs 81.8 kilograms, and has a graduate degree!**

**JASON WILDER (single) was born in 2000, weighs 81.8 kilograms, and has a graduate degree!**

**jason wilder (single) was born in 2000, weighs 81.8 kilograms, and has a graduate degree!**

**JASON WILDER (single) was born in 2000, weighs 180.0 pounds, and has a graduate degree!**

**jason wilder (single) was born in 2000, weighs 180.0 pounds, and has a graduate degree!**

**Santa Claus (married) was born in 1000, weighs 280.0 pounds, and has a high school diploma!**

**Santa Claus (married) was born in 1000, weighs 127.3 kilograms, and has a high school diploma!**

**SANTA CLAUS (married) was born in 1000, weighs 127.3 kilograms, and has a high school diploma!**

**santa claus (married) was born in 1000, weighs 127.3 kilograms, and has a high school diploma!**

**SANTA CLAUS (married) was born in 1000, weighs 280.0 pounds, and has a high school diploma!**

**santa claus (married) was born in 1000, weighs 280.0 pounds, and has a high school diploma!**