Upload your assignment to the Learning Hub at learn.bcit.ca before the start of lesson 12 into the Activities🡪Assignments🡪Assignment3 dropbox.

Ensure that every method and function is named as a verb, has DocString comments, follows good indentation and other style guidelines.

Create three python files named Country.py, data.py, and main.py, with the functionality as follows. The bottom of every script must have the following guard code as the final two lines:

if \_\_name\_\_ == '\_\_main\_\_':

main()

Country.py

This file contains the Country class. Define a constructor that takes three parameters: a country name, a capital name, and a population. In the constructor, assign the parameter values to the instance variables, unless the population is less than 2 million; in that case raise a ValueError with a message in the format of (“Population 1057988 is too low”).

Also add the following methods:

print\_details() This method prints data in this format:

"The capital of Canada (pop. 1234567) is OTTAWA"

birth() This method adds 1 to the object’s own self’s population

death() This method subtracts 1 from the object’s own self’s population

disaster() For countries with a population of 100 million or higher, this method subtracts 100 million from the population. For smaller countries, it sets the population to 0.

# data.py

This script has a main function that prints “I should not be called”…the main function does nothing else.

This script also creates and populates same countries\_and\_capitals tuple that we used in assignment 1. You can copy/paste the code from right here into your data.py script:

# START HERE AND PASTE INTO YOUR LAB IN YOUR IDE

countries\_and\_capitals = (['Afghanistan', 'Kabul'], ['Albania', 'Tirana (Tirane)'], ['Algeria', 'Algiers'], ['Andorra', 'Andorra la Vella'], ['Angola', 'Luanda'], ['Antigua and Barbuda', "Saint John's"], ['Argentina', 'Buenos Aires'], ['Armenia', 'Yerevan'], ['Australia', 'Canberra'], ['Austria', 'Vienna'], ['Azerbaijan', 'Baku'], ['Bahamas', 'Nassau'], ['Bahrain', 'Manama'], ['Bangladesh', 'Dhaka'], ['Barbados', 'Bridgetown'], ['Belarus', 'Minsk'], ['Belgium', 'Brussels'], ['Belize', 'Belmopan'], ['Benin', 'Porto Novo'], ['Bhutan', 'Thimphu'], ['Bolivia', 'Sucre'], ['Bosnia and Herzegovina', 'Sarajevo'], ['Botswana', 'Gaborone'], ['Brazil', 'Brasilia'], ['Brunei', 'Bandar Seri Begawan'], ['Bulgaria', 'Sofia'], ['Burkina Faso', 'Ouagadougou'], ['Burundi', 'Gitega'], ['Cambodia', 'Phnom Penh'], ['Cameroon', 'Yaounde'], ['Canada', 'Ottawa'], ['Cape Verde', 'Praia'], ['Central African Republic', 'Bangui'], ['Chad', "N'Djamena"], ['Chile', 'Santiago'], ['China', 'Beijing'], ['Colombia', 'Bogota'], ['Comoros', 'Moroni'], ['Congo, Democratic Republic of the', 'Kinshasa'], ['Congo, Republic of the', 'Brazzaville'], ['Costa Rica', 'San Jose'], ["Cote d'Ivoire (Ivory Coast)", 'Yamoussoukro'], ['Croatia', 'Zagreb'], ['Cuba', 'Havana'], ['Cyprus', 'Nicosia'], ['Czech Republic (Czechia)', 'Prague'], ['Denmark', 'Copenhagen'], ['Djibouti', 'Djibouti'], ['Dominica', 'Roseau'], ['Dominican Republic', 'Santo Domingo'], ['East Timor', 'Dili'], ['Ecuador', 'Quito'], ['Egypt', 'Cairo'], ['El Salvador', 'San Salvador'], ['England', 'London'], ['Equatorial Guinea', 'Malabo'], ['Eritrea', 'Asmara'], ['Estonia', 'Tallinn'], ['Eswatini (Swaziland)', 'Mbabana'], ['Ethiopia', 'Addis Ababa'], ['Federated States of Micronesia', 'Palikir'], ['Fiji', 'Suva'], ['Finland', 'Helsinki'], ['France', 'Paris'], ['Gabon', 'Libreville'], ['Gambia', 'Banjul'], ['Georgia', 'Tbilisi'], ['Germany', 'Berlin'], ['Ghana', 'Accra'], ['Greece', 'Athens'], ['Grenada', "Saint George's"], ['Guatemala', 'Guatemala City'], ['Guinea', 'Conakry'], ['Guinea-Bissau', 'Bissau'], ['Guyana', 'Georgetown'], ['Haiti', 'Port au Prince'], ['Honduras', 'Tegucigalpa'], ['Hungary', 'Budapest'], ['Iceland', 'Reykjavik'], ['India', 'New Delhi'], ['Indonesia', 'Jakarta'], ['Iran', 'Tehran'], ['Iraq', 'Baghdad'], ['Ireland', 'Dublin'], ['Israel', 'Jerusalem'], ['Italy', 'Rome'], ['Jamaica', 'Kingston'], ['Japan', 'Tokyo'], ['Jordan', 'Amman'], ['Kazakhstan', 'Nur-Sultan'], ['Kenya', 'Nairobi'], ['Kiribati', 'Tarawa Atoll'], ['Kosovo', 'Pristina'], ['Kuwait', 'Kuwait City'], ['Kyrgyzstan', 'Bishkek'], ['Laos', 'Vientiane'], ['Latvia', 'Riga'], ['Lebanon', 'Beirut'], ['Lesotho', 'Maseru'], ['Liberia', 'Monrovia'], ['Libya', 'Tripoli'], ['Liechtenstein', 'Vaduz'], ['Lithuania', 'Vilnius'], ['Luxembourg', 'Luxembourg'], ['Madagascar', 'Antananarivo'], ['Malawi', 'Lilongwe'], ['Malaysia', 'Kuala Lumpur'], ['Maldives', 'Male'], ['Mali', 'Bamako'], ['Malta', 'Valletta'], ['Marshall Islands', 'Majuro'], ['Mauritania', 'Nouakchott'], ['Mauritius', 'Port Louis'], ['Mexico', 'Mexico City'], ['Moldova', 'Chisinau'], ['Monaco', 'Monaco'], ['Mongolia', 'Ulaanbaatar'], ['Montenegro', 'Podgorica'], ['Morocco', 'Rabat'], ['Mozambique', 'Maputo'], ['Myanmar (Burma)', 'Nay Pyi Taw'], ['Namibia', 'Windhoek'], ['Nauru', 'No official capital'], ['Nepal', 'Kathmandu'], ['Netherlands', 'Amsterdam'], ['New Zealand', 'Wellington'], ['Nicaragua', 'Managua'], ['Niger', 'Niamey'], ['Nigeria', 'Abuja'], ['North Korea', 'Pyongyang'], ['North Macedonia (Macedonia)', 'Skopje'], ['Northern Ireland', 'Belfast'], ['Norway', 'Oslo'], ['Oman', 'Muscat'], ['Pakistan', 'Islamabad'], ['Palau', 'Melekeok'], ['Panama', 'Panama City'], ['Papua New Guinea', 'Port Moresby'], ['Paraguay', 'Asuncion'], ['Peru', 'Lima'], ['Philippines', 'Manila'], ['Poland', 'Warsaw'], ['Portugal', 'Lisbon'], ['Qatar', 'Doha'], ['Romania', 'Bucharest'], ['Russia', 'Moscow'], ['Rwanda', 'Kigali'], ['Saint Kitts and Nevis', 'Basseterre'], ['Saint Lucia', 'Castries'], ['Saint Vincent and the Grenadines', 'Kingstown'], ['Samoa', 'Apia'], ['San Marino', 'San Marino'], ['Sao Tome and Principe', 'Sao Tome'], ['Saudi Arabia', 'Riyadh'], ['Scotland', 'Edinburgh'], ['Senegal', 'Dakar'], ['Serbia', 'Belgrade'], ['Seychelles', 'Victoria'], ['Sierra Leone', 'Freetown'], ['Singapore', 'Singapore'], ['Slovakia', 'Bratislava'], ['Slovenia', 'Ljubljana'], ['Solomon Islands', 'Honiara'], ['Somalia', 'Mogadishu'], ['South Africa', 'Pretoria, Bloemfontein, Cape Town'], ['South Korea', 'Seoul'], ['South Sudan', 'Juba'], ['Spain', 'Madrid'], ['Sri Lanka', 'Colombo'], ['Sudan', 'Khartoum'], ['Suriname', 'Paramaribo'], ['Sweden', 'Stockholm'], ['Switzerland', 'Bern'], ['Syria', 'Damascus'], ['Taiwan', 'Taipei'], ['Tajikistan', 'Dushanbe'], ['Tanzania', 'Dodoma'], ['Thailand', 'Bangkok'], ['Togo', 'Lome'], ['Tonga', "Nuku'alofa"], ['Trinidad and Tobago', 'Port of Spain'], ['Tunisia', 'Tunis'], ['Turkey', 'Ankara'], ['Turkmenistan', 'Ashgabat'], ['Tuvalu', 'Funafuti'], ['Uganda', 'Kampala'], ['Ukraine', 'Kiev'], ['United Arab Emirates', 'Abu Dhabi'], ['United Kingdom', 'London'], ['United States', 'Washington D.C.'], ['Uruguay', 'Montevideo'], ['Uzbekistan', 'Tashkent'], ['Vanuatu', 'Port Vila'], ['Vatican City', 'Vatican City'], ['Venezuela', 'Caracas'], ['Vietnam', 'Hanoi'], ['Wales', 'Cardiff'], ['Yemen', "Sana'a"], ['Zambia', 'Lusaka'], ['Zimbabwe', 'Harare'])

# END HERE

# main.py

This script imports data.py and its variables, as follows:

**from data import countries\_and\_capitals**

The main function **try**’s to create a list of Country objects called all\_countries (assigning each one a random population between 1 million and 1 billion, in the try block) using the countries\_and\_capitals tuple (use a loop); make sure any ValueError exceptions are handled by printing “oops” to the console in addition to the ValueError’s message of “Population 1057988 is too low”.

If there are no exceptions, loop through the all\_countries list and call each object’s print\_details() method.

Then loop through the list again, calling birth() for each object and calling print\_details() again; note that each country’s population should be 1 higher than the previous loop.

Then loop through the list again, calling death() for each object and calling print\_details() again; note that each country’s population should be 1 lower than the previous loop.

Then loop through the list again, calling disaster() for each object and calling print\_details() again; note that each country’s population should be 100 million lower than the previous loop (for big countries) or zero (for countries that previously had populations of 100 million or lower).