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 🡪 Lab1a) before the start of lesson 2.

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 Harrison \*/).

Below is a Python class. It defines properties and methods for bank accounts. Another file has a main function which creates some bank account objects and calls their methods. Create the corresponding Java bank account class and main class (which contains the main method) that does the same things. Comment every class, constructor, and method. Use proper naming conventions, indentation, etc…

**bank\_account.py**

class BankAccount:

def \_\_init\_\_(self, balance\_cdn, account\_number, member\_last\_name):

self.balance\_cdn = balance\_cdn

self.account\_number = account\_number

self.member\_last\_name = member\_last\_name

def withdraw(self, amount\_cdn):

self.balance\_cdn -= amount\_cdn

def deposit(self, amount\_cdn):

self.balance\_cdn += amount\_cdn

def transfer(self, amount\_cdn, recipient\_account):

recipient\_account.deposit(amount\_cdn)

self.withdraw(amount\_cdn)

**main.py**

from bank\_account import BankAccount

def main():

b1 = BankAccount(100.00, "abc123", "gates")

b2 = BankAccount(500.00, "xyz789", "woods")

print(b1.balance\_cdn)

b1.withdraw(5.00)

print(b1.account\_number)

print(b1.balance\_cdn)

print("---")

print(b2.balance\_cdn)

b2.deposit(23.00)

print(b2.member\_last\_name)

print(b2.balance\_cdn)

print("---")

b1.transfer(50.00, b2)

print(b1.balance\_cdn)

print(b2.balance\_cdn)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**output**

100.0

abc123

95.0

---

500.0

woods

523.0

---

45.0

573.0