**package** ECTE331\_ProjB;

**class** ProjB {

**private** **static** **int** *A1*, *A2*, *A3*, *B1*, *B2*, *B3*;

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

Object lock = **new** Object();

Thread ThreadA = **new** Thread(**new** Runnable() {

@Override

**public** **void** run() {

**synchronized** (lock) {

*A1* = Utility.*sum*(100);

lock.notify(); // A1 then proceeds to B1 in ThreadB after it is finished calculating

**try** {

lock.wait(); // Here it is waiting for B2 to finish calculating from ThreadB in order for A2 to be next

*A2* = *B2* + Utility.*sum*(400);

lock.notify(); // It then notifies ThreadB to switch the order from A2 to B3

lock.wait(); // Here A3 is waiting for B3 to finish calculating in order to finish the sequence as shown in the figure

*A3* = *B3* + Utility.*sum*(600);

lock.notify(); // Checks if there is any other task left at ThreadB else it will continue the program

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

}

});

Thread ThreadB = **new** Thread(**new** Runnable() {

@Override

**public** **void** run() {

**synchronized** (lock) {

**try** {

lock.wait(); // Here, B1 is waiting for A1 to finish calculating from ThreadA

*B1* = *A1* + Utility.*sum*(200);

// At this line, the figure displays that it goes from B1 to B2 so no notify or wait is needed

*B2* = Utility.*sum*(300);

lock.notify(); // The program then switches back to ThreadA to calculate A2.

lock.wait(); // Wait for ThreadA to finish A2 calculation

*B3* = *A2* + Utility.*sum*(500);

lock.notify(); // Notifies A3 to proceed its calculation (as the figure 2.1 says)

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

}

});

ThreadA.start();

ThreadB.start();

**try** {

ThreadA.join(); // waits for both of the threads to finish before compiling the next set of codes below

ThreadB.join(); // this means that the code is technically paused until the Threads A and B are finished (which then continues after)

} **catch** (InterruptedException e) {

e.printStackTrace();

}

//Printf statements in order:

System.***out***.println("Final Values:");

System.***out***.println("A1: " + *A1*);

System.***out***.println("A2: " + *A2*);

System.***out***.println("A3: " + *A3*);

System.***out***.println("B1: " + *B1*);

System.***out***.println("B2: " + *B2*);

System.***out***.println("B3: " + *B3*);

//Printf statements in the order of the FIGURE 2.1

System.***out***.println("\nFinal Values as shown in the Figure 2.1:");

System.***out***.println("A1: " + *A1*);

System.***out***.println("B1: " + *B1*);

System.***out***.println("B2: " + *B2*);

System.***out***.println("A2: " + *A2*);

System.***out***.println("B3: " + *B3*);

System.***out***.println("A3: " + *A3*);

}

}

//the sum method is set in a dedicated utility class

**class** Utility **extends** ProjB {

// the sum calculation as shown in the appendix (word file)

**public** **static** **int** sum(**int** n) {

**return** n \* (n + 1) / 2;

}

}