***** PROJECT 0B Hardcopy (pdf file) *****

********** Cover page **********

Class: 323 MW

Name: Adewole Adeoshun

Project: Project 0B

Project name:C++ project submission exercise

Language: C++

Due date: 8/31/2025, Sunday before midnight, 11:59pm

Submit date: 9/9/2025

Top level algorithm steps

Step 0: inFile ← open from argv [1]
outFile ← open from argv [2]

Step 1: numOfRows ← read from inFile.

Step 2: Persons people[] ← new Person[numOfRows]; // create an array of Persons;

Step 3: index ← 0 // set initial counter to 0

Step 4: name ← read from inFile.

Step 5: age ← read from inFile

Step 6: p ← new Person (name, age) // create the Person object

Step 7: people[index++] = p; // save the person

Step 8: repeat steps 4 to 7 while index < numOfRows

Step 9: For each person in people array, print the person.

Step 10: delete all allocations

Step 11: close inFile, outFile.

Illustration:

None

Source code

```
#include <iostream>

#include <fstream>

#include <string>

using namespace std;

// Class holding name and age.

class Person {

private:

string name;

int age;

public:

// Constructor

Person(string name, int age) {

this->name = name;

this->age = age;

}

void printPerson(ofstream &ofile) {

ofile << name << " is " << age << " years old.\n";

}//end print

}; // end class Person
```

```
int main(int argc, char** argv) {

if (argc != 3) {

cout << "Your command line need to include two parameters: input file and output file \n";

exit(1);

} // end if argc

// Open input file from argv[1]

ifstream inFile(argv[1]);

if (!inFile.is_open()) {

cout << "Unable to open the input file" << endl;

exit(1);

}

// Open output file from argv[2]

ofstream outFile(argv[2]);

if (!outFile.is_open()) {

cout << "Unable to open the output file" << endl;

exit(1);

}

int numOfPeople;

inFile >> numOfPeople;

//Write header line before listing people

outFile << "*** There are " << numOfPeople << " people ***\n";

// Create an array of Person pointers

Person** people = new Person*[numOfPeople];

string Tname;
```

```
int Tage;
int index = 0;
// Read each person's name and age, create Person objects, and store pointers
while (index < numOfPeople) {
inFile >> Tname;
inFile >> Tage;
Person* p = new Person(Tname, Tage);
people[index++] = p; // save the person pointer
} // end while()
// Print each person to the output file
for (int index = 0; index < numOfPeople; index++) {
people[index]->printPerson(outFile);
} // end for()
// Clean up dynamically allocated memory and close files
for (int index = 0; index < numOfPeople; index++) {
delete people[index];
}
delete[] people;
inFile.close();
outFile.close();
exit(0);
} // end main()
```

Output

*** There are 7 people ***

Sean is 24 years old.

Pamela is 29 years old.

Mark is 22 years old.

Danial is 17 years old.

John is 18 years old.

Eric is 24 years old.

Jessica is 23 years old.