**ID Lab**

IDdriver.cpp

ID.cpp

ID.h

**Introduction:**

This lab will create a class named ID which will be used by the main program called IDdriver.cpp.

The objective of this lab is to create an object that initializes an 8 digit ID\_number and a 9 character full\_ID that begins with the character ‘A’ (example: “A12345678”). Declare all the methods and constructors (getters, setters, default constructors) in the .h file while defining and implementing them in the .cpp file.

**Instructions:**

In order to create the class, separate the code into 2 files, one named “ID.h” and the other named “ID.cpp”.

**In the .h file**

Declare private members for the ID\_number and full\_ID.

int ID\_number; // e.g. 12345678 (just the integer)

string full\_ID; // includes the “A” e.g. “A12345678” 9 characters in total

Declare member functions, getters and setters, as well as the default constructor.

**In the .cpp file:**

Define the default constructor to:

* Set ID\_number

Hint: use:

static int currentNumber = 10000000;

// increment after setting ID\_number so each subsequent object instance has a new number

* initialize the full\_ID as ‘A’ and the ID\_number .

Note: ID\_number has to be converted to a string in order to concatenate it to the character.

**In main (IDdriver.cpp)**

Dynamically create 5 unique ID objects using a for loop. Append those ID dynamic objects to a vector of pointers to ID objects. Finally, print the IDs in the vector, first to last .

Compile the code with:

g++ IDdriver.cpp ID.cpp

**Example Output:**

A10000001

A10000002

A10000003

A10000004

A10000005