Lab Manual – Classes

Student will learn how to build

- Overloaded constructor
- Copy constructor
- This pointer
- Destructor

Task 1

Implement a class called BiggerInt. The BiggerInt class will have two data members:

- int* big_int_; // Pointer to the int array that holds the big integer i
- int int_length_; // Variable to store the length of the big integer

(While an integer is of 4 bytes in size with a range of -2,147,483,648 to 2,147,483,647. A big integer can store long integer numbers with no size limitation.)

You have to implement the following:

1. Write a default constructor and initialize big_int_ to nullptr.
    - BiggerInt();
2. Write an overloaded constructor and perform copy.
    - BiggerInt (const int * obj, int size);
3. Write a member function to make a deep copy of the big_int_ of the passed BiggerInt obj into the big_int_ of the object which called this function.
    - void assign(const BiggerInt & obj);
4. Write a member function which will overload the above assign function and perform the same operations but the argument passed to this function is a pointer integer array.
    - void assign(const int * big_int, int size);
5. Write a member function to append the big_int_ of the passed BiggerInt obj to the end of big_int_ of the object which called this function.
    - void append(const BiggerInt & obj);
6. Write a member function which will overload the above append function and perform the same operations but the argument passed to this function is a pointer integer array.
    - void append(const int* big_int, int size);
7. Write a member function to compare the big_int_ of BiggerInt obj with the big_int_ of the object which called this function. Return 0 for equal, 1 for less than and 2 for greater than.
    - int compareTo(const BiggerInt & obj);
8. Write a member function which overloads the above compareTo function and performs the same operations but the argument passed to this function is a pointer integer array.
    - int compareTo(const int* big_int, int size);
9. void display();

10. Write a destructor to deallocate any dynamically allocated memory.
    - ~ BiggerInt();
11. Write a suitable main() function to test all the functions of the BiggerInt class.
12. Write a member function to display the big_int_ on screen. If big_int_ is empty, print "No Value Assigned".

Note:

- Deallocate all dynamically allocated memory.
- Make separate my_big_int.h, my_big_int.cpp.
- Do not use any string class built-in functions except for strlen(), if required.
- Follow all the code indentation, naming conventions and code commenting guidelines.

Task 2

Implement a class "Car" that have three data members

- Char *model
- Char *company
- Int year

It has the following member functions.

- Default constructor that initialize null to char* data members and assign "0" to year data members.
    - Car()
- Copy Constructor
- Parameterized Constructor
- Add a new car in inventory.
    - AddCar(const Car & c)
- Update detail of existing car in inventory
    - UpdateCar(const Car &c)

In main the system should provide the following options to the user:

a. Add a new car to the inventory.
b. Update the details of an existing car.
c. Delete a car from the inventory.
d. View the list of available cars.
e. Exit the system.

When the user selects the option to add a new car:

a. The system prompts the user to enter the make, model, year, and other details of the car.
b. The system creates a Car object with the provided details and adds it to the car inventory.

When the user selects the option to update an existing car:

a. The system displays the list of available cars

b. The user selects a car from the list to update.

c. The system prompts the user to enter the updated details of the car.

d. The system updates the car's details in the inventory.

When the user selects the option to view the list of available cars:

a. The system displays the list of cars in the inventory, including their make, model, and year.

When the user selects the option to exit the system:

a. The program terminates, and the car management system closes.