Object Oriented Programming Lab

Lab 03 **Marks 10**

Instructions

- Work in this lab individually.
- You can use your books, notes, handouts etc. but you are not allowed to borrow anything from your peer student.
- Make sure to follow the best coding practices.
- Include comments to explain the logic where necessary.
- Test your program thoroughly with various inputs to ensure proper functionality and error handling.
- Show your work to the instructor before leaving the lab to get some or full credit.

Task 01

1. Declare two integer pointers named **ptrX** and **ptrY** and initialize them with **nullptr**.
2. Create two variables with values 2 and 8 on heap memory segment and assign their addresses to **ptrX** and **ptrY** respectively.

3. Print the following information:
    - The address of **ptrX**, value of **ptrX** and the value of **memory location** where it points to.
    - The address of **ptrY**, value of **ptrY** and the value of **memory location** where it points to.

4. Free the resources allocated on the heap memory segment.

Task 02

1. Ask the user to enter the size of a float array.
2. Allocate memory to an array based on the size provided by the user.
3. Initialize the array content by reading them from user.
4. Calculate and display the average of array elements.
5. Free any memory resources allocated by the program before exit.

Task 03

Implement a function named **getEvenNumbers** with the following signature:

```
int* getEvenNumbers(const int ar[], const int size, int& evenArraySize);
```

The parameters **ar** and **size** holds an array and its size respectively.

The function should return a pointer to newly created array which contains only even numbers exist in array **ar** and store its **size** in parameter **evenArraySize**. It should store 0 (zero) in **evenArraySize** and return **nullptr** if **ar** contains only odd numbers. *The function should not display anything.*

In the **main** function declare an array of size 10. Fill the array with arbitrary values and then pass it to **getEvenNumbers** function along with its size and all the required parameters. Display contents of the array returned by function **getEvenNumbers** if any, otherwise display a message "No Negative Numbers Exist in the Array!". Don't forget to free the memory resource allocated by the program, if any.

Task 04

Implement following function named ***getPosNeg*** that accept an array **ar** along with its size **n_ar**

```
void getPosNeg(const int ar[], const int n_ar, int* &pos, int& n_pos, int* &neg, int& n_neg);
```

The parameters **ar** and **n_ar** hold an array and its size respectively.

The function gets all the positive and negetive numbers from the array **ar** and places them into newly created arrays pointed by parameter **pos** and **neg** respectively. Store the sizes of **pos** and **neg** arrays into **n_pos** and **n_neg** respectively. It should store 0 (zero) and **nullptr** in parameters **n_pos/n_neg** and **pos/neg** respectively, if **ar** has no positive and or negative numbers. *The function should not display anything.*

In the **main** function declare an array of size 10. Fill the array with arbitrary values and then pass it to **getPosNeg** function along with its size and all the required parameters. After the execution of function display contents of the arrays pointed by **pos** and **neg**. Display appropriate message(s), if **pos** and or **neg** arrays are empty. Don't forget to free the memory resource allocated by the program, if any.