

Object Oriented Programming Lab

Lab 01

Marks 05

Instructions

Work on this lab individually. You can use your books, notes, handouts etc. but you are not allowed to borrow anything from your peer student.

Marking Criteria

Show your work to the instructor before leaving the lab to get some or full credit.

What you must do

Program the following tasks in your C++ compiler and then compile and execute them.

Task 1

Write a program that performs the following tasks

1. Declare **float variables** named **f** and **t** and initialize them with **2.5** and **8.9** respectively.
2. Declare **pointer variables** **ptrF** and **ptrT** and initialize them with the **addresses** of **f** and **t** respectively.

Now print the following information:

1. The address of **f** and the value of **f**.
2. The address of **ptrF**, value of **ptrF** and the value of **memory location** where it points to.
3. The address of **t** and the value of **t**.
4. The address of **ptrT**, value of **ptrT** and the value of **memory location** where it points to.

Task 2

Implement the following **function** named **sumByPointer** that accept **three pointers to integer**.

```
void sumByPointer(int* ptrA, int* ptrB, int* ptrR);
```

The function should **calculate the sum** of the **contents of memory locations** pointed by **first two pointers** (*ptrA and ptrB*) and store the **sum** in third (*ptrR*). *The function should not display anything.*

In **main** function **asks the user to input two doubles** and then **passes them to subtractByPointer** function. The **result of subtraction** should be **displayed** on the screen. *The main function should not perform any calculations.*

Task 3

Implement the following **function** named **arrayByPointer** that accept **a pointer to double** and an **integer** to hold the **size**.

```
void arrayByPointer(double* ptrArray, int size);
```

The parameters *ptrArray* and *size* holds **starting address** and the **size** of an **array** respectively.

The function should display **memory address of each array location along with its contents** clearly. You are **not allowed** to use **subscript operator** `[]` in **arrayByPointer** function.

In **main** function declare an **array of size 10**. Fill the array with arbitrary values and then passes it to **arrayByPointer** function along with its **size**.

Task 4

Implement the following **function** named **getEvenOddSum** that accept **a pointer to integer** and an **integer** to hold the **size**. The function accepts **two additional pointers to integers** to store the **sum of even and odd** elements exist in the **array**.

```
void getEvenOddSum(int* ptrArray, int size, int* ptrEvenSum, int* ptrOddSum);
```

The parameters *ptrArray* and *size* holds **starting address** and the **size** of an **array** respectively.

The function calculates the **sum of even and odd** numbers exist in the array pointed by *ptrArray* and places them to the memory locations pointed by parameters **ptrEvenSum** and **ptrOddSum** respectively. *The function should not display anything.*

In **main** function declare an **array of size 10**. Fill the array with arbitrary values and then passes it to **getEvenOddSum** function along with its **size**. Display the **sum of even and odd** numbers clearly on the screen. *The main function should not perform any calculations.*