

## Lab 10 – Pointers

### Task 01:

You're managing student information for a school. Your task is to implement a function that swaps the information of two students. This task will help you understand how pointers can be applied to real-world problems.

#### Code:

```
#include<iostream>
using namespace std;

int main(){
    string name[2], grade[2];
    int age[2];

    string *pName = name, *pGrade = grade;
    int *pAge = age;

    for(int i = 0; i < 2; i++){
        cout << "Student no. " << i+1 << endl;
        cout << "\tEnter name: ";
        getline(cin, name[i]);
        cout << "\tEnter age: ";
        cin >> age[i];
        cout << "\tEnter grade: ";
        cin >> grade[i];
        cin.ignore();
    }

    cout << "Before swaping:" << endl;
    for(int i = 0; i < 2; i++){
        cout << "\tStudent no. " << i+1 << " Name: " << name[i] << " Age: " << age[i] << "
Garde: " << grade[i] << endl;
    }

    //swaping
    string tempName = *pName;
    *pName = *(pName+1);
    *(pName+1) = tempName;

    int tempAge = *pAge;
    *pAge = *(pAge+1);
    *(pAge+1) = tempAge;
```

```
string tempGrade = *pGrade;
*pGrade = *(pGrade+1);
*(pGrade+1) = tempGrade;

cout << "After swaping:" << endl;
for(int i = 0; i < 2; i++){
    cout << "\tStudent no. "<<i+1<<" Name: "<<name[i]<<" Age: "<<age[i]<<"
Garde: "<<grade[i] << endl;
}
}
```

**Output:**

```
PS D:\Hasan\cpp\00. university\lab10-Pointers> g++ task01.cpp
PS D:\Hasan\cpp\00. university\lab10-Pointers> ./a.exe
Student no. 1
    Enter name: Hasan
    Enter age: 18
    Enter grade: 80
Student no. 2
    Enter name: Anas
    Enter age: 19
    Enter grade: 70
Before swaping:
    Student no. 1 Name: Hasan Age: 18 Garde: 80
    Student no. 2 Name: Anas Age: 19 Garde: 70
After swaping:
    Student no. 1 Name: Anas Age: 19 Garde: 70
    Student no. 2 Name: Hasan Age: 18 Garde: 80
PS D:\Hasan\cpp\00. university\lab10-Pointers> █
```

**Task 02:**

Imagine you're managing expenses for a grocery shopping trip with your friends. Your task is to develop a function that computes the total expenditure and average spending per person based on the amounts spent by each individual on groceries. Implement the function to accept an array of integers representing the amounts spent by each person on groceries, using call by reference with pointers.

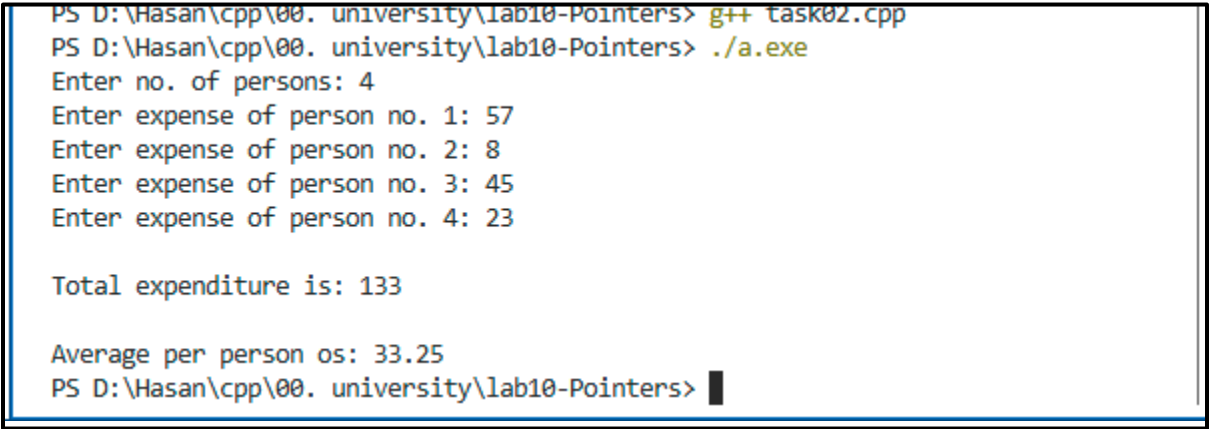
**Code:**

```
#include<iostream>
using namespace std;

void avgAndTotal(int *a, int size){
    int total = 0;
    for(int i = 0; i < size; i++) total += *(a + i);

    float avg = total/float(size);
    cout << "Total expenditure is: " << total << endl;
    cout << "Average per person os: " << avg << endl;
}

int main(){
    int person = 0;
    cout << "Enter no. of persons: ";
    cin >> person;
    int a[person];
    for(int i = 0; i < person; i++){
        cout << "Enter expense of person no. "<<i+1<<": ";
        cin >> a[i];
    }
    avgAndTotal(a, person);
}
```

**Output:**

```
PS D:\Hasan\cpp\00. university\lab10-Pointers> g++ task02.cpp
PS D:\Hasan\cpp\00. university\lab10-Pointers> ./a.exe
Enter no. of persons: 4
Enter expense of person no. 1: 57
Enter expense of person no. 2: 8
Enter expense of person no. 3: 45
Enter expense of person no. 4: 23

Total expenditure is: 133

Average per person os: 33.25
PS D:\Hasan\cpp\00. university\lab10-Pointers> █
```

**Task 03:**

You're tasked with organizing a shelf of books in a library. Your task is to create a function that arranges the books in ascending order based on their heights. You can use pointers to efficiently manage their positions.

**Code:**

```
#include<iostream>
using namespace std;

void organizeBooks(float *a, int size){
    cout << "\nHeights of Books before organizing:" << endl;
    for(int i = 0; i < size; i++){
        cout <<*(a+i)<<"cm ";
    }
    cout << endl;

    //organizing
    for(int i = 0; i < size; i++){
        for(int j = 0; j < (size-i-1); j++){
            if(*(a+j) > *(a+j+1)){
                float temp = *(a+j);
                *(a+j) = *(a+j+1);
                *(a+j+1) = temp;
            }
        }
    }

    cout << "\nHeights of Books after organizing:" << endl;
    for(int i = 0; i < size; i++){
        cout <<*(a+i)<<"cm ";
    }
    cout << endl;
}

int main(){
    int noOfBooks = 0, bookCount = 0, choice = 0;
    cout << "Welcome to Book Shelf Organizer" << endl << endl;
    cout << "Enter the no. of books: ";
    cin >> noOfBooks;

    float shelf[noOfBooks];
    do{
        cout << "\n---MENU---" << endl;
        cout << "1. Add book." << endl;
```

```
    cout << "2. See organized books." << endl;
    cout << "3. Exit." << endl;
    cout << "Enter choice: ";
    cin >> choice;
    if(choice == 1){
        if(bookCount >= noOfBooks){
            cout << "\nBook shelf is full." << endl;
        }
        else{
            cout << "\nEnter height of book (cm): ";
            cin >> shelf[bookCount];
            cout << "Book added!" << endl;
            bookCount++;
        }
    }
    else if(choice == 2){
        organizeBooks(shelf, bookCount);
    }
    else if(choice == 3){
        cout << "\nClosing program." << endl;
    }
    else{
        cout << "\nInvalid choice!" << endl;
    }
}while(choice != 3);
}
```

**Output:**

```
PS D:\Hasan\cpp\00. university\lab10-Pointers> g++ task03.cpp
PS D:\Hasan\cpp\00. university\lab10-Pointers> ./a.exe
Welcome to Book Shelf Organizer

Enter the no. of books: 3

---MENU---
1. Add book.
2. See organized books.
3. Exit.
Enter choice: 1

Enter height of book (cm): 56
Book added!

---MENU---
1. Add book.
2. See organized books.
3. Exit.
Enter choice: 1

Enter height of book (cm): 3.4
Book added!

---MENU---
1. Add book.
2. See organized books.
3. Exit.
Enter choice: 2

Heights of Books before organizing:
56cm 3.4cm

Heights of Books after organizing:
3.4cm 56cm

---MENU---
```

```
---MENU---
```

1. Add book.
2. See organized books.
3. Exit.

```
Enter choice: 1
```

```
Enter height of book (cm): 89
```

```
Book added!
```

```
---MENU---
```

1. Add book.
2. See organized books.
3. Exit.

```
Enter choice: 1
```

```
Book shelf is full.
```

```
---MENU---
```

1. Add book.
2. See organized books.
3. Exit.

```
Enter choice: 2
```

```
Heights of Books before organizing:
```

```
3.4cm 56cm 89cm
```

```
Heights of Books after organizing:
```

```
3.4cm 56cm 89cm
```

```
---MENU---
```

1. Add book.
2. See organized books.
3. Exit.

```
Enter choice: 3
```

```
Closing program.
```