

LAB 1

REVIEW ON POINTERS + CLASSES

Data Structures
2020-2021

AGENDA

- Review on Pointers:
 - Declarations and Definition
 - Arrays and Pointers
 - New and Delete operators
- Classes:
 - Exercise: Student Class

10 minutes

TASK 1

What will be the output of the following code?

```
int count = 10, *temp, sum = 0;
temp = &count;
*temp = 20;
temp = &sum;
*temp = count;
count++;
(*temp)--;
printf("count = %d, *temp = %d, sum = %d\n",
count, *temp, sum );
```

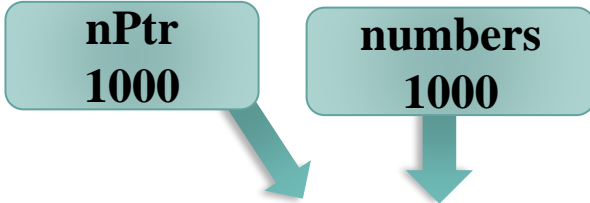
```
count = 21, *temp = 19, sum = 19
Press any key to continue . . .
```

Task 2

10 minutes

- What will be the output in each?

```
for (int i = 0; i < 10; i++)
{
    // pointer/subscript notation
    cout << numbers[i];
    cout << nPtr [i];
    // pointer/offset notation
    cout << *(nPtr + i) ;
    cout << *(numbers+i) << endl;
}
```



Address	Variable	Memory
1000	numbers [0]	0.0
1008	numbers [1]	1.1
1016	numbers [2]	2.2
1024	numbers [3]	3.3
1032	numbers [4]	4.4
...
1072	numbers [9]	9.9
996	⁵ nPtr	1000

TASK 3

```
int *zPtr; // zPtr will reference array z
int number;
int z[ 5 ] = { 1, 2, 3, 4, 5 };
```

Find the error in the following:

```
++zPtr;
```

Error: zPtr has not been initialized.

Correction: Initialize zPtr first with zPtr = z;

TASK 3

```
int *zPtr; // zPtr will reference array z
int number;
int z[ 5 ] = { 1, 2, 3, 4, 5 };
```

Find the error in the following:

```
// use pointer to get first value of array
number = zPtr;
```

Error: The pointer is not dereferenced.

Correction: Change the statement to `number = *zPtr;`

TASK 3

```
int *zPtr; // zPtr will reference array z
int number;
int z[ 5 ] = { 1, 2, 3, 4, 5 };
```

Find the error in the following:

```
// assign array element 2 (the value 3) to number
number = *zPtr[2];
```

Error: zPtr[2] is not a pointer and should not be dereferenced.

Correction: Change *zPtr[2] to zPtr[2].

TASK 3

```
int *zPtr; // zPtr will reference array z
int number;
int z[ 5 ] = { 1, 2, 3, 4, 5 };
```

Find the error in the following:

```
++z;
```

Error: Trying to modify an array name with pointer arithmetic.

Correction: Use a pointer variable instead of the array name to accomplish pointer arithmetic, or subscript the array name to refer to a specific element.

15 minutes

TASK 4

Trace the following code segment:

```
//prototype of function copyArray
void copyArray(int *, int *, int);

void main () {
    int size = 5;
    int* originalArr =
        new int [size];

    cout<<"Enter the array:\n";
    for (int i = 0; i < size; i++)
    {
        cin>>originalArr[i];
    }

    int *copiedArr = new int[size];
    copyArray(originalArr,
        copiedArr, size);
    cout<<"The copied array:\n";
```

```
for (int i = 0; i < size; i++)
{
    cout<<copiedArr[i]<<" ";
}
delete [] originalArr;
delete [] copiedArr;
}
```

```
// the function copyArray
void copyArray(int * originalArr, int *
copiedArr , int size) {
    for (int i = 0; i < size; i++) {
        copiedArr[i] = originalArr[i];
    }
}
```

Enter the array values:

1
2
3
4
5

The copied array:

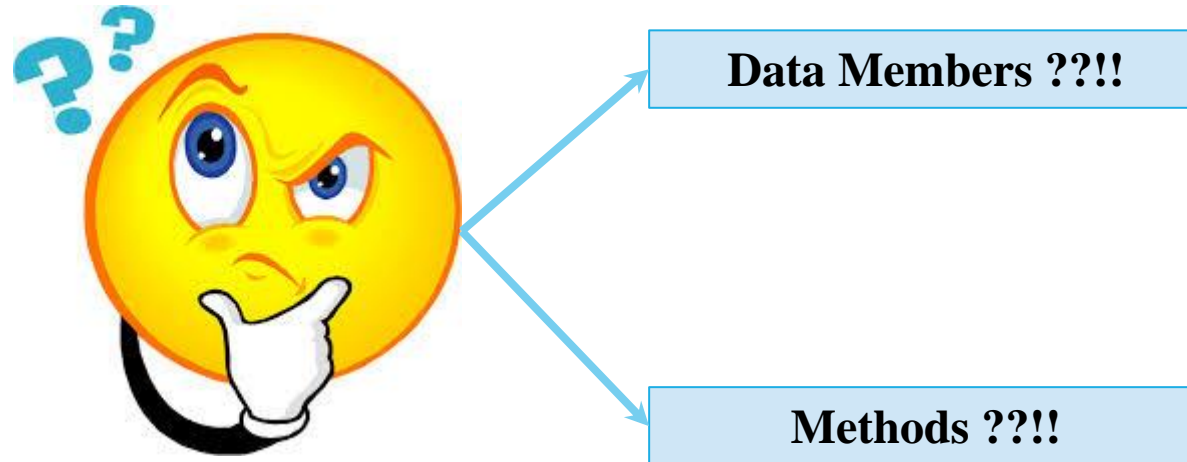
1 2 3 4 5

Press any key to continue . . .

STUDENT CLASS EXAMPLE-TASK 5

Assume each student in school has an ID and 3 marks for 3 different subjects.

Student affairs unit want to a program that allow them to enter the data for any number of students and calculate the total marks for each student then display the data of all students.



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

```
class Student{
```

```
    int ID;
    double marks[3];
    int total;
```

Data Members

```
public:
```

```
    Student()
    {
        total = 0;
    }
    Student(int std_ID)
    {
        ID = std_ID;
        total = 0;
    }
```

Constructors

Methods

```
void readStudentData()
{
    cout<<"ID: ";
    cin>>ID;
    for (int i = 0; i < 3; i++) {
        cout<<"Subject "<<i+1<<": ";
        cin>>marks[i];
    }
}
void sumStudentsMarks()
{
    for (int i = 0; i < 3; i++) {
        total+=marks[i];
    }
}
};
```

```
#include "Student.h"
```

```
int main()
```

```
{
```

```
    int studsNum;
```

```
    cout<<"Enter the number of students:";
```

```
    cin>>studsNum;
```

```
    Student * studs = new Student[studsNum];
```

```
    // read the students' data; id & 3 grades
```

```
    for (int i = 0; i < studsNum; i++) {
```

```
        cout<<"Student :"<< i+1<<endl;
```

```
        studs[i].readStudentData();
```

```
        studs[i].sumStudentsMarks();
```

```
    }
```

```
    // display the students' data; id & 3 grades
```

```
    cout<<"*****Displaying:*****\n";
```

```
    for (int i = 0; i < studsNum; i++) {
```

```
        cout<<"Student :"<< i+1<<endl;
```

```
        studs[i].displayStudentData();
```

```
    }
```

```
}
```

MAIN FILE

STUDENT CLASS IN 2 SEPARATE FILES

.H FILE

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

class Student{
    int ID;
    double marks[3];
    int total;
public:
    Student();
    Student(int std_ID);
    void readStudentData() ;
    void sumStudentsMarks();
};
```

.CPP FILE

```
#include "Student.h"
Student::Student()
{
    total=0;
}
Student::Student(int std_ID)
{
    ID = std_ID;
    total=0;
}
void Student::readStudentData(){
    cout<<"ID: ";
    cin>>ID;
    for (int i = 0; i < 3; i++) {
        cout<<"Subject "<<i+1<<": ";
        cin>>marks[i];
    }
}
void Student::sumStudentsMarks(){
    for (int i = 0; i < 3; i++) {
        total+=marks[i];
    }
}
```

Thank You