



CL-1004 Object Oriented Programming Lab No 6

Objectives:

- Classes and objects
- Data members
- Member function
- Data encapsulation
- Member access specifier (private , public)
- Constructor, Destructor

Note: Carefully read the following instructions (*Each instruction contains a weightage*)

1. There must be a block of comments at start of every question's code by students; the block should contain brief description about functionality of code.
2. Comment on every function about its functionality.
3. Use understandable name of variables.
4. Proper indentation of code is essential
5. Write a C++ statement(s) for each of the following task one after the other, in the same order.
6. Make a Microsoft Word file and paste all of your C++ code with all possible screenshots of every **task output in MS word and submit .cpp file with word file.**
7. Make separate .cpp files for all tasks and use this format **22F-1234_Task1.cpp**.
8. First think about statement problems and then write/draw your logic on copy.
9. After copy pencil work, code the problem statement on MS Studio C++ compiler.
10. At the end when you done your tasks, attached C++ created files in MS word file and make your submission on Google classroom. (Make sure your submission is completed).
11. Please submit your file in this format **22F-1234_L1**.
12. Do not submit your assignment **after deadline**.
- 13. Do not copy code from any source otherwise you will be penalized with negative marks.**



Problem 1: | : (Classes, objects, Constructor, Destructor and Member functions, constant)

Write a class Employee with following data members and methods.

Private:

1. ID //A string to hold the ID of employee.
2. Name //A string to hold the name of employee.
3. Department //A string to hold the department name of employee.
4. Bank Account number //A string to hold the organizational bank account number of employee (A read-only data member)
5. Grade //A string to hold the employee grade (1 to 8, 8 being the highest) of employee.

Public:

1. **Default Constructor** should initialize all member variables with nullvalue.
2. **Overloaded Constructor** should initialize all (initialize-able) data members of the employee class.
3. **inputData** Input from user for all data members
4. **setter** set data of all attributes.
5. **getter** get data of all attributes.
6. **displayData** method that display data using (member function)
7. **Destructor**

Now create 3 other objects and initialize them with user input data. Show your results on console.

Also dynamically create 3 other objects and initialize them with user input data. Show your results on console.

Explicitly call **destructors** for all **local** objects with proper message.

Problem 2: | (Classes, objects, Constructor, Destructor and Member functions, constant, copy constructor)

Write a class Person, having following private data members:

1. Name
2. Data of birth (a constant data member)
3. CNIC (a constant data member)
4. Marital Status

Public member functions:

1. Constant member function to access the Date of Birth of each person (get DoB)
2. Constant member function to access the CNIC of each person (get CNIC)
3. Display function for Person record output (const)

Display a person record Mr. X with DoB 1st January 2001 having CNIC xxxxx-xxxxxxx-x



Problem: 3 |

Find the error in each of the following segments. If the error can be corrected, explain how.

- a) `int *number;`
`cout << number << endl;`
- b) `double *realPtr;`
`long *integerPtr;`
`integerPtr = realPtr;`
- c) `int * x, y;`
`x = y;`
- d) `char s[] = "this is a character array";`
`for (;*s != '\0'; ++s)`
`cout << *s << ' ';`
- e) `short *numPtr,`
`result; void *genericPtr = numPtr;`
`result = *genericPtr + 7;`
- f) `double x = 19.34;`
`double xPtr = &x;`
`cout << xPtr << endl;`

Problem: 4 | HW

What does these programs do?

Part a)

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

void mystery1(char*, const char*); // prototype

int main()
{
    char string1[80];
    char string2[80];

    cout << "Enter two strings: ";
    cin >> string1 >> string2;
    mystery1(string1, string2);
    cout << string1 << endl;
} // end main

// What does this function do?
void mystery1(char* s1, const char* s2)
{
    while (*s1 != '\0')
        ++s1;

    for (; *s1 = *s2; ++s1, ++s2)
        ; // empty statement
} // end function mystery1
```



Part b)

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

int mystery2(const char*); // prototype
int main()
{
    char string1[80];

    cout << "Enter a string: ";
    cin >> string1;
    cout << mystery2(string1) << endl;
} // end main

// What does this function do?
int mystery2(const char* s)
{
    int x;
    for (x = 0; *s != '\0'; ++s)
        ++x;
    return x;
} // end function mystery2
```

Problem: 5 | HW

Explain the prefix and postfix pointers with proper and reasonable comments on each statement about the working of pointers.

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int a = 5, b = 10;
    int c;
    int* p1, * p2;
    p1 = &a;
    p2 = &b;
    c = *p1;
    cout << "*(p1++) =" << *(p1++) << endl;
    cout << "value of p1 " << p1 << endl;
    cout << "*(++p1) =" << *(++p1) << endl;
    cout << "value of p1 " << p1 << endl;
    cout << "(*p1)++ =" << (*p1)++ << endl;
    cout << "value of p1 " << p1 << endl;
    cout << "++(*p1) =" << ++(*p1) << endl;
    cout << "value of p1 " << p1 << endl;
    return 0;
}
```

Problem: 6 | HW

Write a program which declares a variable of character type named alpha and assign it the value "Y". Now perform the tasks mentioned below:



1. Declare a **pointer pointing to the constant character** and pass the address of alpha in it. Now change the value of alpha to Z using this pointer. See what output you expect?
2. Declare a **constant pointer to the character** and pass the address of alpha in it. Now declare another character variable named bravo and assign it value M. Now try to pass address of bravo to the same constant pointer and see what happens?
3. Declare a **constant pointer to the constant character** and pass the address of alpha in it. Now declare another character variable named bravo and assign it value M. Now try to pass address bravo to the same constant pointer .Try to modify the value of alpha too and also see what happens?

Proper code indentation will hold extra marks!

Best of luck 😊

You are done with your exercise, submit on classroom at given time



National University



Of Computer & Emerging Sciences Faisalabad-Chiniot Campus
