

Lab Manual

**CS214L – Object Oriented Programming
Lab**

Lab No: 06

Topic: This pointer, Pointers and Objects

Class: BSGM

Semester: II

Session: Spring, 2022

Lab Date: March 15th, 2022

Lab Time: 11:40hrs – 2:30hrs



Air University Islamabad
FACULTY OF COMPUTING & ARTIFICIAL
INTELLIGENCE

Department of Creative Technologies

Instructions

Submission: Use proper naming convention for your submission file. Name the submission file as LabNO_ROLLNUM (e.g. Lab01_00000). Submit the file on Google Classroom within the deadline. Failure to submit according to the above format would result in deduction of 10% marks. Submissions on the email will not be accepted.

Plagiarism: Plagiarism cases will be dealt with strictly. If found plagiarized, both the involved parties will be awarded zero marks in the assignment, all of the remaining assignments, or even an F grade in the course. Copying from the internet is the easiest way to get caught!

Deadline: The deadlines to submit the assignment are hard. Late submission with marks deduction will be accepted according to the course policy shared by the instructor. Correct and timely submission of the assignment is the responsibility of every student; hence no relaxation will be given to anyone.

Comments: Comment your code properly. Bonus marks (maximum 10%) will be awarded to well comment code. Write your name and roll number (as a block comment) at the beginning of the solution to each problem.

Tip: For timely completion of the assignment, start as early as possible. Furthermore, work smartly - as some of the problems can be solved using smarter logic.

1. Note: Follow the given instructions to the letter, failing to do so will result in a zero.



Objectives

In this lab, you will learn:

- Accessing Member data with “this” pointer
- Array of objects
- Pointers and Objects

Concepts

Every object has access to its **own address** through a **pointer** called **this** (a C++ keyword). Thus any member function can find out the **address of the object** of which it is a member.

1. Array of Objects:

As with any other data type in C++, you can define **arrays of class objects**. An array of **InventoryItem** objects could be created to represent a business’s inventory records. Here is an example of such a definition:

```
const ARRAY_SIZE = 10;
InventoryItem inventory[ARRAY_SIZE];
```

This statement defines an array of 40 **InventoryItem** objects. The name of the array is **inventory**, and the default constructor is called for each **object in the array**.

3.1 Accessing Members of Objects in an Array:

Objects in an array are accessed with **subscripts**, just like any other data type in an array. For example, to call the **setUnits member function** of **inventory[2]**, the following statement could be used:

```
inventory[2].setUnits(30);
```

This statement sets the units variable of **inventory[2]** to the value 30.

2. Pointers and Objects:

4.1 Pointer to Object:

Both the pointer and the object are **writable**. You can modify the object, e.g. changing its **x value** and you can also modify the pointer, e.g. **assign it a new object**.

```
InventoryItem item;
InventoryItem item2;
InventoryItem* inventory_ptr = &item;

inventory_ptr = &item2;    // Modify pointer, OK
```



```
inventory_ptr->setUnits(20); // Modify object, OK
```

4.2 Const Pointer to Object

Here we can't modify the pointer but **object properties can be modified.**

```
InventoryItem item;
InventoryItem item2;
InventoryItem* const inventory_ptr = &item;

inventory_ptr = &item2;    // Modify pointer, ERROR
inventory_ptr->setUnits(20); // Modify object, OK
```

4.3 Pointer to Const Object

Here we can modify the pointer but **we can't modify the object.**

```
InventoryItem item;
InventoryItem item2;
const InventoryItem* inventory_ptr = &item;

inventory_ptr = &item2;    // Modify pointer, OK
inventory_ptr->setUnits(20); // Modify object, ERROR
```

4.4 Const Pointer to Const Object

We can't do anything here, **except for reading the object value.**

```
InventoryItem item;
InventoryItem item2;
const InventoryItem* const inventory_ptr = &item;

inventory_ptr = &item2;    // Modify pointer, ERROR
inventory_ptr->setUnits(20); // Modify object, ERROR
```



Demo

This demo implements the complete program that demonstrates the *InventoryItem* class. This program demonstrates “this pointer”, constobject, array of objects, and pointer to objects.

```
#include<iostream>
#include<string>
#include<iomanip>
usingnamespacestd;

classInventoryItem
{
private:
    string description; // The item description
    int units; // Number of units on hand
public:
    InventoryItem()
    { // Assign values to description, cost, and units.
        description = "";
        units = 0;
    }
    // Constructor #3 Parameterized
    InventoryItem(string desc, intu)
    { // Assign values to description, cost, and units.
        description =desc;
        units = u;
    }

    // Mutator functions
    void setDescription(string description)
    {
        this->description =description; // using this pointer
    }

    void setUnits(int units)
    {
        this->units = units; //using this pointer
    }

    // Accessor functions
    string getDescription()
    {
        return description;
    }

    int getUnits() const
    {
        return units;
    }
};

int main()
{
    // array of Objects
    const int ARRAY_SIZE = 4;
    InventoryItem inventory[ARRAY_SIZE];
```



```

inventory[2].setUnits(30); // for setting object values
cout<< inventory[2].getUnits(); // for getting object values

// pointer to object

InventoryItem item;
InventoryItem item2;
InventoryItem* inventory_ptr = &item;

inventory_ptr = &item2;          // Modify pointer, OK
inventory_ptr->setUnits(20);      // Modify object, OK

// const pointer to object

InventoryItem* const inventory_constPtr = &item;

inventory_constPtr = &item2;      // Modify pointer, ERROR
inventory_constPtr->setUnits(20);  // Modify object, OK

// pointer to const object

const InventoryItem* inventory_ptr_constObj = &item;

inventory_ptr_constObj = &item2;    // Modify pointer, OK
inventory_ptr_constObj->setUnits(20); // Modify object, ERROR

// const pointer to const object

Const InventoryItem* const inventory_constPtr_constObj = &item;

inventory_constPtr_constObj = &item2;    // Modify pointer, ERROR
inventory_constPtr_constObj->setUnits(20); // Modify object, ERROR

return 0;
}

```



Lab Tasks

1. Define a C++ class to represent a Student. The attributes are: id, name, number of subjects, marks_per_subject. The number of subjects varies for each student. Note: Use pointers where needed. Accept data for 'n' students and print the mark list for each student on the screen.
 2. Create a class staff having fields: Staff_id , name, salary. Write a menu driven program for:
 - To accept the data
 - To display the data
 - To sort the data by name
 3. Write a C++ program to find the area of circle using class circle which have following details:
 - Accept radius from the user and while calculating area, get value of a radius from a pointer not directly from the object
 - Calculate the area
 - Display the resultApply the following concepts:
 - Pointer to Object
 - Constant pointer to object
 - Pointer to constant object
-

