

Assignment- 2.

I. Choose the correct option :-

- Q1) Local variables are stored in an area called Stack.
- (a) Heap
 - (c) Free memory
 - (b) Permanent
 - (d) Stack.

Q2) Choose the correct option?

#include <iostream.h>

namespace std;

class Base { };

class Derived : public

Base { };

int main()

{

Base * bp = new

Derived;

Derived * dp = new

Base; }

(a) No Compiler Error

(b) Compiler Error in line "Base * bp =
new derived;"

(c) Compiler Error in line "Derived
* dp = new Base";

(d) Runtime Error

- Q3) When the inheritance is private, the private methods in base class are inaccessible in the derived class (in C++).

- (a) Inaccessible
- (b) Accessible

(c) Protected

(d) Public

Q4) Which of the following is true?

- (a) The number of times destructor is called depends on number of objects created.
- (b) Destructor is called only once.
- (c) There can be more than one destructors in the class.
- (d) Programmer have always call destructor at the end of the program.

Q5) State true or false.

Type Conversion is automatic whereas type Casting is explicit.

- (a) True
- (b) False

II: Short Answer type Question

Q1) Explain about new and delete keywords with code.

Ans → Syntax to use new operator :- To allocate memory of any data type, the syntax is :
Pointer - variable = new data-type ;

delete operator is used to deallocate the memory. User has privilege to deallocate the created pointer variable by this delete operator.

Syntax :

// Release memory allocation pointed by pointer-variable

delete pointer-variable;

lets took a example :-

Allocation and deallocation of memory using new and delete.

```
#include <bits/stdc++.h>
```

Using namespace std;

```
int main()
```

```
{
```

```
    int *P = NULL;
```

```
    P = new (nothrow) int;
```

```
    if (!P)
```

```
        Cout << "allocation of the memory failed \n";
```

```
    else
```

```
        *P = 89;
```

```
    Cout << "Value of P: " << *P << endl;
```

```
}
```

```
float *x = new float (12.86);
```

```
Cout << "value of x: " << *x << endl;
```

```
int n = 8;
```

```
int *q = new (nothrow) int [n];
```

```
if (!q)
```

```
Cout << "allocation on the memory failed \n";
```

```
else {
```

```
    for (int i = 0; Re = i < n; i++)
```

```
        q[i] = i + 1;
```

```
Cout << "Value store in block of memory: ";
```

```
for (int i = 0; i < n; i++)
```

```
Cout << q[i] << " "; }
```

O/P:- Value of P: 89

```
delete P;
```

Value of x: 12.89

```
delete x;
```

Value Store in block of memory:

```
delete [] q;
```

1 2 3 4 5 6 7 8

```
} return 0;
```

//new & delete operator:-

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists several projects and files under 'assignment2'. The file 'new_delete.cpp' is currently selected.
- Code Editor:** The main area displays the code for 'new_delete.cpp'. The code demonstrates dynamic memory allocation using 'new' and deallocation using 'delete'. It also prints memory values and stores them in an array.
- Terminal:** A terminal window titled 'Command Prompt' is open, showing the execution of the program. The output shows the allocation of memory, the value of pointer p (89), the value of pointer r (12.86), and the stored values in array q (1 2 3 4 5 6 7 8).
- Bottom Status Bar:** Shows file statistics (Ln 10, Col 9), tab size (Tab Size: 4), encoding (UTF-8), file type (C++), and operating system (Win32).
- Taskbar:** At the bottom, the taskbar includes icons for File Explorer, Task View, Mail (25 notifications), File Explorer, and the Visual Studio Code icon.

```
#include <bits/stdc++.h>
using namespace std;

int main ()
{
    int* p = NULL;
    p = new(nothrow) int;
    if (!p)
        cout << "allocation of memory failed\n";
    else
    {
        *p = 89;
        cout << "Value of p: " << *p << endl;
    }
    float *r = new float(12.86);

    cout << "Value of r: " << *r << endl;
    int n = 8;
    int *q = new(nothrow) int[n];

    if (!q)
        cout << "allocation of memory failed\n";
    else
    {
        for (int i = 0; i < n; i++)
            q[i] = i+1;

        cout << "Value store in block of memory: ";
        for (int i = 0; i < n; i++)
            cout << q[i] << " ";
    }
    delete p;
    delete r;
    delete[] q;

    return 0;
}
```

Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\hp>cd C:\Users\hp\Desktop\c++lpu\assignment2
C:\Users\hp\Desktop\c++lpu\assignment2>g++ new_delete.cpp
C:\Users\hp\Desktop\c++lpu\assignment2>a
Value of p: 89
Value of r: 12.86
Value store in block of memory: 1 2 3 4 5 6 7 8
C:\Users\hp\Desktop\c++lpu\assignment2>

Q2) What are Constructors? Why they are required?
Explain different types of Constructors with
Suitable example.

Ans → A Constructor is a member function of a class which initializes objects of a class. In C++ Constructor is automatically called when object (instance of class) create. It is special member function of class. The main purpose of the class Constructor in C++ programming is to construct an object of a class. In other word it is used to initialize all class data members.

Types of Constructors :-

1. Default Constructor :- Default Constructor is the Constructor which doesn't take any argument.

It has no parameter.

Syntax :- class-name (parameter1, parameter2,)
{ // constructor definition
 }

Example :-

```
#include <bits/stdc++.h>
```

```
Using namespace std;
```

```
Class Public :
```

```
Class Cube
```

```
{
```

```
Public
```

```
int Side;
```

```
Cube ()
```

```
{ Side = 10;
```

```
    }
```

```
int main ()
```

{

Cube c;

cout << c.side;
return 0;

O/P → 10

2) Parameterized Constructors :- These are the Constructors with parameters. Using this constructor you can provide different values to data members of different Objects, by passing the appropriate values as argument.

Example :-

```
#include <bits/stdc++.h>
Using namespace std;
class Cube
{
public:
    int side;
    Cube (int x)
    {
        side = x;
    }
};

int main()
{
    Cube c1(10);
    Cube c2(20);
    Cube c3(30);
    cout << c1.side;
    cout << c2.side;
    cout << c3.side;
    return 0;
}
```

Output :- 10

20

30

//default constructor:-

File Edit Selection View Go Run Terminal Help

default.cpp - c++lpu - Visual Studio Code

Getting Started hello.cpp new_delete.cpp default.cpp X sort.cpp class.cpp variables.cpp question2.cpp \ function.cpp

assignment2 > default.cpp > main()

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 class cube
4 {
5     public:
6         int side;
7         cube()
8     {
9             side=10;
10        }
11    };
12    int main()
13    {
14        cube c;
15        cout<<c.side;
16
17    }
18
19
20
```

Command Prompt

```
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Users\hp\Desktop\c++lpu\assignment2

C:\Users\hp\Desktop\c++lpu\assignment2>g++ default.cpp

C:\Users\hp\Desktop\c++lpu\assignment2>a
10
C:\Users\hp\Desktop\c++lpu\assignment2>
```

Ln 14, Col 16 Spaces: 4 UTF-8 CRLF C++ Win32

Type here to search

4:04 PM 6/15/2021

//parameterized constructor:-

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists files and folders. The file `para.cpp` is selected.
- Code Editor:** The main area displays the code for `para.cpp`. The code defines a class `Cube` with a parameterized constructor that initializes the member variable `side`. The `main` function creates three `Cube` objects with side lengths of 10, 20, and 30, and prints each side length to the console.

```
#include<bits/stdc++.h>
using namespace std;
class Cube
{
public:
    int side;
    Cube(int x)
    {
        side=x;
    }
    int main()
    {
        Cube c1(10);
        Cube c2(20);
        Cube c3(30);
        cout << c1.side << endl;
        cout << c2.side << endl;
        cout << c3.side << endl;
        return 0;
    }
}
```

- Terminal:** A terminal window titled "Command Prompt" shows the execution of the code. It navigates to the directory `C:\Users\hp\Desktop\c++lpu\assignment2`, runs `g++ para.cpp`, and then executes the resulting binary `a`. The output is 10, 20, and 30, corresponding to the sides of the cubes.

```
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Users\hp\Desktop\c++lpu\assignment2
C:\Users\hp\Desktop\c++lpu\assignment2>g++ para.cpp
C:\Users\hp\Desktop\c++lpu\assignment2>a
10
20
30
```

- Bottom Status Bar:** Shows the current line (Ln 16, Col 17), spaces used (Spaces: 4), encoding (UTF-8), file type (C++), and window title (Win32).
- Taskbar:** At the bottom, the taskbar includes icons for File Explorer, Task View, Mail (25 unread), File Explorer, Google Chrome, and Visual Studio Code.
- System Tray:** On the far right of the taskbar, it shows the date (6/15/2021), time (4:08 PM), battery status, signal strength, and a small message icon.

3) Copy Constructor :- It is used to create a copy of an already existing object of a class type. It is usually of the form $X(X\&)$, where X is the class name. The compiler provides a default Copy Constructor to all the classes.

Syntax :-

classname (const classname & object name)
{

}

Example :-

```
#include <bits/stdc++.h>
using namespace std;
class Sample copy constructor
{
```

Private :

```
int x, y;
```

Public :

```
Sample copy constructor (int x1, int y1)
{
```

```
    x = x1;
```

```
    y = y1;
```

```
} Sample copy constructor (const Sample copy constructor & Sam)
{
```

```
    x = Sam.x;
```

```
    y = Sam.y;
```

```
} void display ()
```

```
    cout << x << " " << y << endl;
```

```
}
```

//copy constructor:-

File Edit Selection View Go Run Terminal Help

copy.cpp - c++lpu - Visual Studio Code

Getting Started hello.cpp new_delete.cpp default.cpp para.cpp copy.cpp sort.cpp class.cpp variables.cpp question2.c

assignment2 > copy.cpp > ...

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 class Samplecopyconstructor
4 {
5     private:
6         int x, y;
7     public:
8     Samplecopyconstructor(int x1, int y1)
9     {
10         x = x1;
11         y = y1;
12     }
13     Samplecopyconstructor (const Samplecopyconstructor &sam)
14     {
15         x = sam.x;
16         y = sam.y;
17     }
18     void display()
19     {
20         cout<<x<<" "<<y<<endl;
21     }
22 };
23 int main()
24 {
25     Samplecopyconstructor obj1(10, 15);      // Normal constructor
26     Samplecopyconstructor obj2 = obj1;        // Copy constructor
27     cout<<"Normal constructor : ";
28     obj1.display();
29     cout<<"Copy constructor : ";
30     obj2.display();
31     return 0;
32 }
```

Command Prompt

```
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hp\cd C:\Users\hp\Desktop\c++lpu\assignment2
C:\Users\hp\Desktop\c++lpu\assignment2>a
Normal constructor : 10 15
Copy constructor : 10 15
C:\Users\hp\Desktop\c++lpu\assignment2>
```

Ln 22, Col 3 Spaces: 4 UTF-8 CRLF C++ Win32

Type here to search

int main()
{

 SampleCopyconstructor obj1(10, 15);

 Samplecopyconstructor obj2 = obj1;

 cout << "Normal Constructor : ";

 obj1.display();

 cout << "Copy Constructor : ";

 obj2.display();

 return 0;

}

Output :-

Copy Constructor : 10 15

Q3) Explain the difference between object oriented and procedural programming language in detail.

Ans → Object-Oriented programming Procedural Oriented programming

(i) In OOP, program is divided into small parts called objects. In procedural programming, program is divided into small parts called functions.

(ii) It follows bottom up approach.

It follows top-down approach

(iii) It can access specifier like public, private and protected etc. There is no access specifier in procedural programming.

Object-Oriented Programming

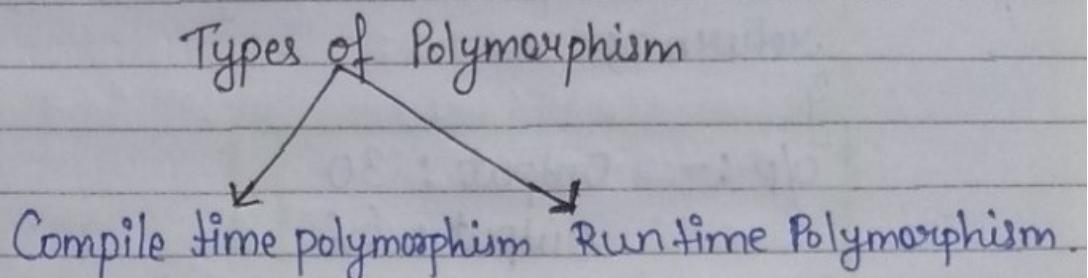
Procedural Programming

- (iv) Adding new data and function is easy.
- (v) Object Oriented programming provides data hiding so it is more secure.
- (vi) Overloading is possible in object - Oriented programming.
- (vii) In object-Oriented programming data is more important than function.
- (viii) It is based on real world.
- (ix) Examples :- C++, Java, Python, C# etc.
- Adding new data and function is not easy.
- In procedural programming, overloading is not possible.
- In procedural programming function is more important than data.
- It is based on unreal world.
- Examples:- C, FORTAN, Pascal, Basic. etc.

III. Long Answer type question :-

Q1) Explain the type of Polymorphism with code:-

Ans → Polymorphism → The word polymorphism means having many forms. Polymorphism means that a call to a member function will cause a different function to be executed depending on the type of object invokes the function.



(i) Compile Time Polymorphism :— This is also known as Static (or early) binding. function Overloading and operator Overloading are perfect example of Compile time polymorphism.
 Example :— In this example we have two functions with same name but different number of arguments. Based on how many parameters we pass during function call determines which function is to be called, this is why it is considered as an example of polymorphism because in different conditions the output is different. Since, the call is determined during Compile Time, that's why it is called Compile time polymorphism.

Code :— #include <bits/stdc++.h>

Using namespace std ;
 class Add {

//compile time polymorphism:-

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists files and folders. Under "assignment2", "compile.cpp" is selected. Other files include "hello.cpp", "new_delete.cpp", "default.cpp", "para.cpp", "copy.cpp", "sort.cpp", "class.cpp", and "variables.cpp".
- Code Editor:** The main editor area contains the following C++ code:

```
#include <bits/stdc++.h>
using namespace std;
class Add {
public:
    int sum(int num1, int num2){
        return num1+num2;
    }
    int sum(int num1, int num2, int num3){
        return num1+num2+num3;
    }
};
int main() {
    Add obj;
    cout<<"Output: "<<obj.sum(10, 20)<<endl;
    cout<<"Output: "<<obj.sum(11, 22, 33);
    return 0;
}
```
- Terminal:** A floating terminal window titled "Select Command Prompt" shows the command-line session:

```
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Users\hp\Desktop\c++lpu\assignment2

C:\Users\hp\Desktop\c++lpu\assignment2>g++ compile.cpp

C:\Users\hp\Desktop\c++lpu\assignment2>a
Output: 30
Output: 66
C:\Users\hp\Desktop\c++lpu\assignment2>
```
- Taskbar:** At the bottom, the taskbar shows the Windows Start button, a search bar with "Type here to search", and pinned icons for File Explorer, Mail (25), Photos, and File Explorer.
- Status Bar:** The status bar at the bottom right displays "Ln 1, Col 24" and other system information like "Spaces: 4", "UTF-8", "CRLF", "C++", "Win32", and the date/time "4:23 PM 6/15/2021".

Public :

```
int sum (int num1, int num2) {  
    return num1 + num2;  
}  
  
int sum (int num1, int num2, int num3) {  
    return num1 + num2 + num3;  
}  
}  
  
int main () {  
    Add obj;  
    cout << "Output : " << obj.sum (10, 20) << endl;  
    cout << "Output : " << obj.sum (11, 22, 33);  
    return 0;  
}
```

O/P :- Output : 30

Output : 66

(ii) Runtime Polymorphism :— ~~function This is also~~
This is also known as dynamic (or late) binding.

Example :— Function Overriding is an example of Runtime polymorphism.

Function Overriding : When child class declares a method which is already present in the parent class then this is called function overriding, here child class overrides the parent class. In case of function overriding we have two definitions of same function, one is parent class and one is child class. The call to the function is determined at runtime. To decide which definition of the function is to be called, that's the reason it is called runtime poly.

Code :- #include <bits/stdc++.h>
Using namespace std;
class A {
public :
 void display() {
 cout << "Super class Function" << endl;
 }
};
class B : public A {
public :
 void disp() {
 cout << "Subclass function";
 }
};
int main() {
 A obj;
 obj.display();
 B obj2;
 obj2.disp();
 return 0;
}

O/P :- Superclass function
Subclass Function

//runtime polymorphism:-

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists project files including assignment2, assignment1, assignment2, and runtime.cpp.
- Code Editor:** The main editor shows the content of `runtime.cpp`. The code defines a base class `A` with a virtual function `disp()` and a derived class `B` that overrides it. In `main()`, objects of both classes are created and their `disp()` methods are called.
- Terminal:** A Command Prompt window shows the terminal session:

```
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Users\hp\Desktop\c++lpu\assignment2
C:\Users\hp\Desktop\c++lpu\assignment2>g++ runtime.cpp
C:\Users\hp\Desktop\c++lpu\assignment2>a
Super Class Function
Sub Class Function
C:\Users\hp\Desktop\c++lpu\assignment2>
```
- Bottom Status Bar:** Shows file statistics (0△0), encoding (UTF-8), and system information (Ln 1, Col 24, 4:26 PM, 6/15/2021).
- Taskbar:** At the bottom, the taskbar includes icons for File Explorer, Task View, Mail (25 unread), File Explorer, Google Chrome, and File Explorer.

Q2) Write a program to sort an array of 0,1,2 in the best possible time and space complexity.

For example :-

Input Array :- 112200212

Output Array :- 001112222

Ans → Code :-

```
#include <bits/stdc++.h>
using namespace std;
Void sort 012 (int a[], int arr-size)
{
    int lo=0;
    int hi = arr-size-1;
    int mid = 0;
    while (mid <= hi) {
        switch (a[mid]) {
            Case 0:
                Swap (a[lo++], a[mid++]);
                break;
            Case 1:
                mid++;
                break;
            Case 2:
                Swap (a[mid], a[hi--]);
                break;
        }
    }
}
```

Void Print Array (int arr[], int arr-size)

```
{  
    for (int i=0; i<arr-size; i++)  
        cout << arr[i] << " ";
```

}

// array after segmentation:-

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists several projects and files. Projects include "assignment1", "assignment2", and "Lec1". Files under "assignment2" include "a.exe", "class.cpp", "compile.cpp", "copy.cpp", "default.cpp", "new_delete.cpp", "para.cpp", "runtime.cpp", and "sort.cpp".
- Code Editor:** The main editor area displays the content of "sort.cpp". The code implements a quicksort algorithm for segregating an array into two parts: 0s and 1s.
- Terminal:** A terminal window titled "Command Prompt" is open, showing the command-line interface for running the program. It includes the path "C:\Users\hp\Desktop\c++lpu\assignment2", the compilation command "g++ sort.cpp", and the output of the program, which is "array after segregation : 0 0 1 1 1 2 2 2".
- Bottom Status Bar:** The status bar at the bottom provides information about the current line (Ln 13, Col 1), tab size (Tab Size: 4), encoding (UTF-8), file type (C++), and operating system (Win32).
- Taskbar:** At the very bottom, the Windows taskbar is visible with icons for File Explorer, Mail, File Explorer (with 25 items), and the Visual Studio Code icon.

```
int main()
```

```
{  
    int arr [] = {1, 1, 2, 2, 20, 0, 2, 1, 2};  
    int n = sizeof(arr) / sizeof(arr[0]);  
    Sort012(arr, n);  
    cout << "array after segregation:" ;  
    PrintArray(arr, n);  
    return 0;  
}
```

O/P → array after segregation: 001112222

Q3) Create a class named 'Member' having the following members:

Data members

1- Name	4- Address
2- Age	5- Salary

3- Phone Number

It also has a method named 'Print Salary' which prints the salary of members. Two classes 'Employee' and 'manager' inherit the 'member' class. The 'Employee' and 'manager' classes have data members 'specilization' and 'department' respectively. Now assign name, age, phone no., address and salary to an employee and a manager by making an object of both these classes and print the same.

Ans → Code :- #include <bits/stdc++.h>
Using namespace std;

Class member {

 char name[40], address[50];
 int ~~int~~ number;

 int age;

 Public:

 int salary;

 Void input();

}

 Cout << "Name : " << endl;

 Cin >> name;

 Cout << "Age : " << endl;

 Cin >> age;

 Cout << "Phone number : " << endl;

 Cin >> number;

 Cout << "Address : " << endl;

 Cin >> address;

 Cout << "Salary:" << endl;

 Cin >> Salary;

g

 Void display()

{

 Cout << endl;

 Cout << "Name:" << name << endl;

 Cout << "Age:" << age << endl;

 Cout << "Phone number : " << number << endl;

 Cout << "Address:" << address << endl;

 Cout << "Salary :" << salary << endl;

};

Class employee : Public member
char Specialization [30], department [20];
Public :

Void input ()

{

Cout << "In Enter employee Details:";

member :: input ();

Cout << "Specialization :" << endl;

Cin >> Specialization ;

Cout << "Department :" << endl;

Cin >> department ;

}

Void display ()

{

Cout << "In Displaying Employee Details \n";

member :: display ();

Cout << "Specialization :" << Specialization << endl;

Cout << "Department :" << department << endl;

}

Void PrintSalary ()

{

Cout << "In salary of the member is :" << Salary << endl;

}

Class manager : Public member

char Specialization [30], department [20];

Public :

Cout << "In Enter manager Details \n";

member :: input ();

Cin >> Cout << "Specialization :" << endl;

Cin >> Specialization ;

```
Cout << "Department : " << endl;
cin >> department;
}
```

```
Void display()
```

```
Cout << "In Displaying manager details \n";
member :: display();
```

```
Cout << "Specialization : " << specialization << endl;
Cout << "Department : " << department << endl;
}
```

```
Void printsalary()
```

```
Cout << "Salary of the member is : " << salary << endl;
}
}
```

```
int main()
{
```

```
employee e;
```

```
manager m;
```

```
e.input();
```

```
e.display();
```

```
e.printsalary();
```

```
m.display();
```

```
m.printsalary();
```

```
return 0;
}
```

O/P :- Enter Employee Details :

Name : John

Age : 21

Phone Number : 0123456789

Address : Punjab

Salary : 12000000

Specialization : developer

Department : CSE

Enter Manager Details :

Name : Lariaib

Age : 28

Phone Number : 012345678

Address : Bgp

Salary : 1280000

Specialization : Coder

Department : CSE

Displaying Employee Details :

Name : John

Age : 21

Phone number : 0123456789

Address : Punjab

Salary : 12000000

Specialization : developer

Department : CSE

Salary of the member is : 12000000

Displaying Manager Details :

Name : Lariaib

Age : 28

Phone Number : 012345678

Address : Bgp

Salary : 1280000

Specialization : Coder

Department : CSE

Salary of the Member is : 1280000

//inheritance:-

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists project files and folders. The file `class.cpp` is currently selected.
- Code Editor:** The main editor area contains C++ code demonstrating inheritance. It defines a base class `member` with private members `name`, `address`, `number`, and `age`. It also has public member functions `input()` and `display()`. A derived class `employee` inherits from `member` and adds its own private members: `specialization` and `department`. It overrides the `input()` function to prompt for employee-specific details like specialization and department.
- Terminal:** The right side shows the terminal window with the following output:

```
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Users\hp\Desktop\c++lpu\assignment2
C:\Users\hp\Desktop\c++lpu\assignment2>g++ class.cpp
C:\Users\hp\Desktop\c++lpu\assignment2>a

Enter Employee Details:
Name :
john
Age :
21
Phone Number :
0123456789
Address :
punjab
Salary :
12000000
Specialization :
developer
Department :
cse

Enter Manager Details
Name :
laraib
Age :
```
- Bottom Status Bar:** Shows the current line (Ln 44), column (Col 41), and encoding (UTF-8). It also includes icons for spaces, tabs, and file formats.
- Taskbar:** At the bottom, the taskbar shows the VS Code icon and other open application icons.

File Edit Selection View Go Run Terminal Help

class.cpp - c++lpu - Visual Studio Code

Getting Started hello.cpp new_delete.cpp default.cpp para.cpp copy.cpp compile.cpp runtime.cpp sort.cpp class.cpp

> .vscode
assignment-1
 a.exe
question1.cpp
question2.cpp
question3.cpp
question4.cpp
question5.cpp
assignment2
 a.exe
class.cpp
compile.cpp
copy.cpp
default.cpp
new_delete.cpp
para.cpp
runtime.cpp
sort.cpp
> Lec1
 a.exe
hello.cpp
variables.cpp

assignment2 > class.cpp > employee > display()

```
37 cout<<"Specialization : "<<endl;
38 cin>>specialization;
39 cout<<"Department : "<<endl;
40 cin>>department;
41 }
42 void display()
43 {
44     cout<<"\n Displaying Employee Details \n";
45     member::display();
46     cout<<"Specialization : "<<specialization<<endl;
47     cout<<"Department : "<<department<<endl;
48 }
49 void printSalary()
50 {
51     cout<<"\n Salary of the member is : "<<salary<<endl;
52 }
53 };
54 class manager : public member{
55     char specialization[30], department[20];
56 public:
57     void input()
58     {
59         cout<<"\n \t Enter Manager Details \t \n";
60         member::input();
61         cout<<"Specialization : "<<endl;
62         cin>>specialization;
63         cout<<"Department : "<<endl;
64         cin>>department;
65     }
66     void display()
67     {
68         cout<<"\n \t Displaying Manager Details \t \n";
69         member::display();
70         cout<<"Specialization : "<<specialization<<endl;
71         cout<<"Department : "<<department<<endl;
72     }
73     void printSalary()
```

Command Prompt

Enter Manager Details

Name : laraib
Age : 28
Phone Number : 012345678
Address : bgp
Salary : 1280000
Specialization : coder
Department : cse

Displaying Employee Details

Name : john
Age : 21
Phone Number : 123456789
Address : punjab
Salary : 12000000
Specialization : developer
Department : cse

Salary of the member is : 12000000

Ln 44, Col 41 Spaces: 2 UTF-8 CRLF C++ Win32

Type here to search

File Edit Selection View Go Run Terminal Help class.cpp - c++lpu - Visual Studio Code

Getting Started hello.cpp new_delete.cpp default.cpp para.cpp copy.cpp compile.cpp runtime.cpp sort.cpp class.cpp X

> .vscode
assignment-1
 a.exe
 question1.cpp
 question2.cpp
 question3.cpp
 question4.cpp
 question5.cpp
assignment2
 a.exe
 class.cpp
 compile.cpp
 copy.cpp
 default.cpp
 new_delete.cpp
 para.cpp
 runtime.cpp
 sort.cpp
> Lec1
 a.exe
 hello.cpp
 variables.cpp

59 {
60 cout<<"\n \t Enter Manager Details \t \n";
61 member::input();
62 cout<<"Specialization : "<<endl;
63 cin>>specialization;
64 cout<<"Department : "<<endl;
65 cin>>department;
66 }
67 void display()
68 {
69 cout<<"\n \t Displaying Manager Details \t \n";
70 member::display();
71 cout<<"Specialization : "<<specialization<<endl;
72 cout<<"Department : "<<department<<endl;
73 }
74 void printSalary()
75 {
76 cout<<"\n Salary of the member is : "<<salary<<endl;
77 }
78 };
79 int main()
80 {
81 employee e;
82 manager m;
83 e.input();
84 m.input();
85 e.display();
86 e.printSalary();
87 m.display();
88 m.printSalary();
89
90 return 0;
91 }

1280000
Specialization :
coder
Department :
cse

Displaying Employee Details

Name : john
Age : 21
Phone Number : 123456789
Address : punjab
Salary : 12000000
Specialization : developer
Department : cse

Salary of the member is : 12000000

Displaying Manager Details

Name : laraib
Age : 28
Phone Number : 12345678
Address : bgp
Salary : 1280000
Specialization : coder
Department : cse

Salary of the member is : 1280000

Ln 44, Col 41 Spaces: 2 UTF-8 CRLF C++ Win32 445 PM 6/15/2021

Type here to search