MCQ.

1) Local variable are stored in an area called _____.

a) Heap (c) Free Memory
b) Permanent (d) Stack.

2. Choose the correct option?

```
#include using 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

3) When the inheritance is private, the private methods in base class are _____ in the derived class (in C++).

a. Inaccessible b. Accessible c. Protected
d. Public.

4) 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 to always call destructor at the end of the program.

5.) State true or false

Type conversion is automatic whereas type casting is explicit

A. True B. False

II. Short answer type question.

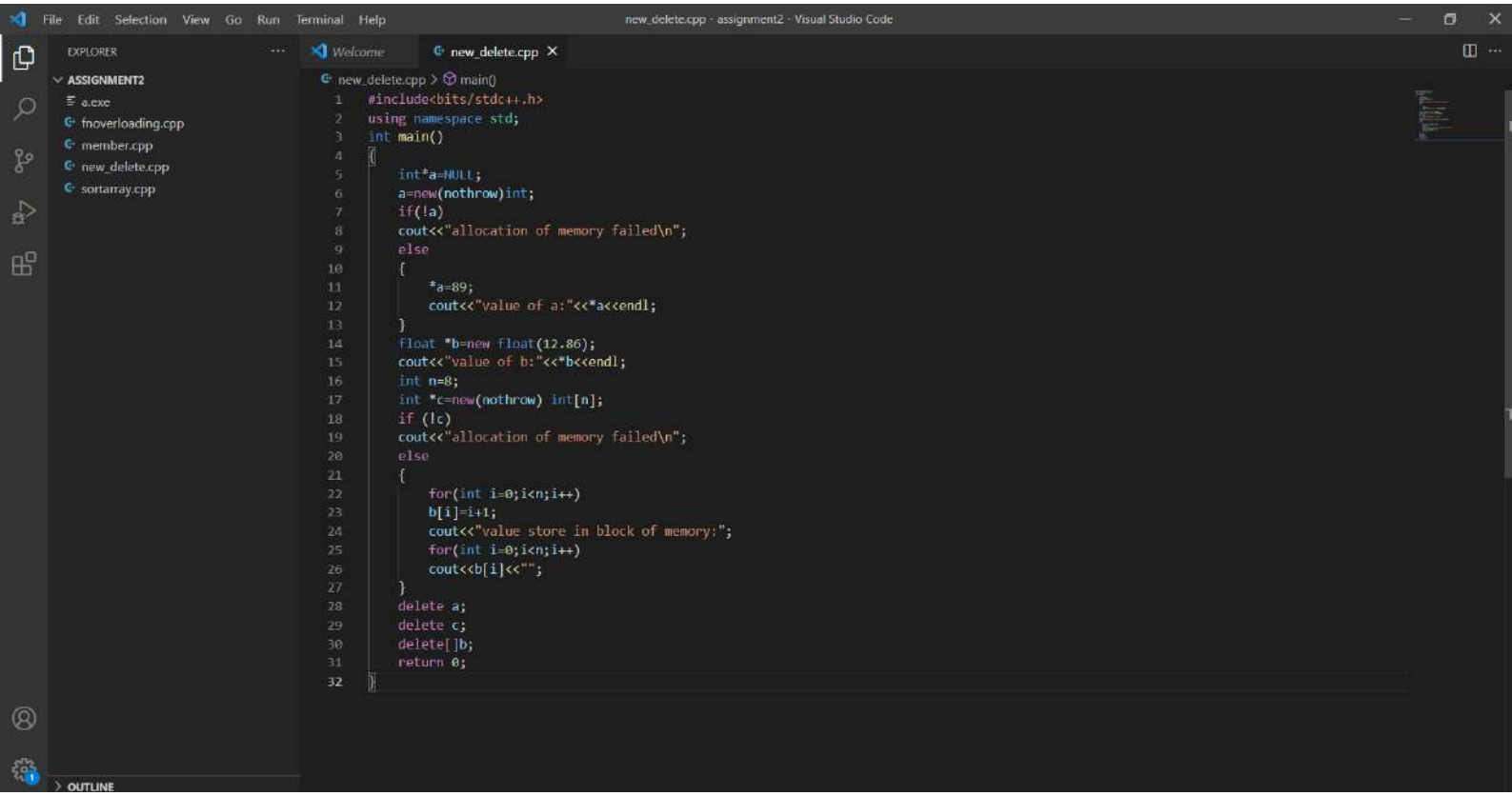
Q.1-> Explain about new and delete Keywords with Code.

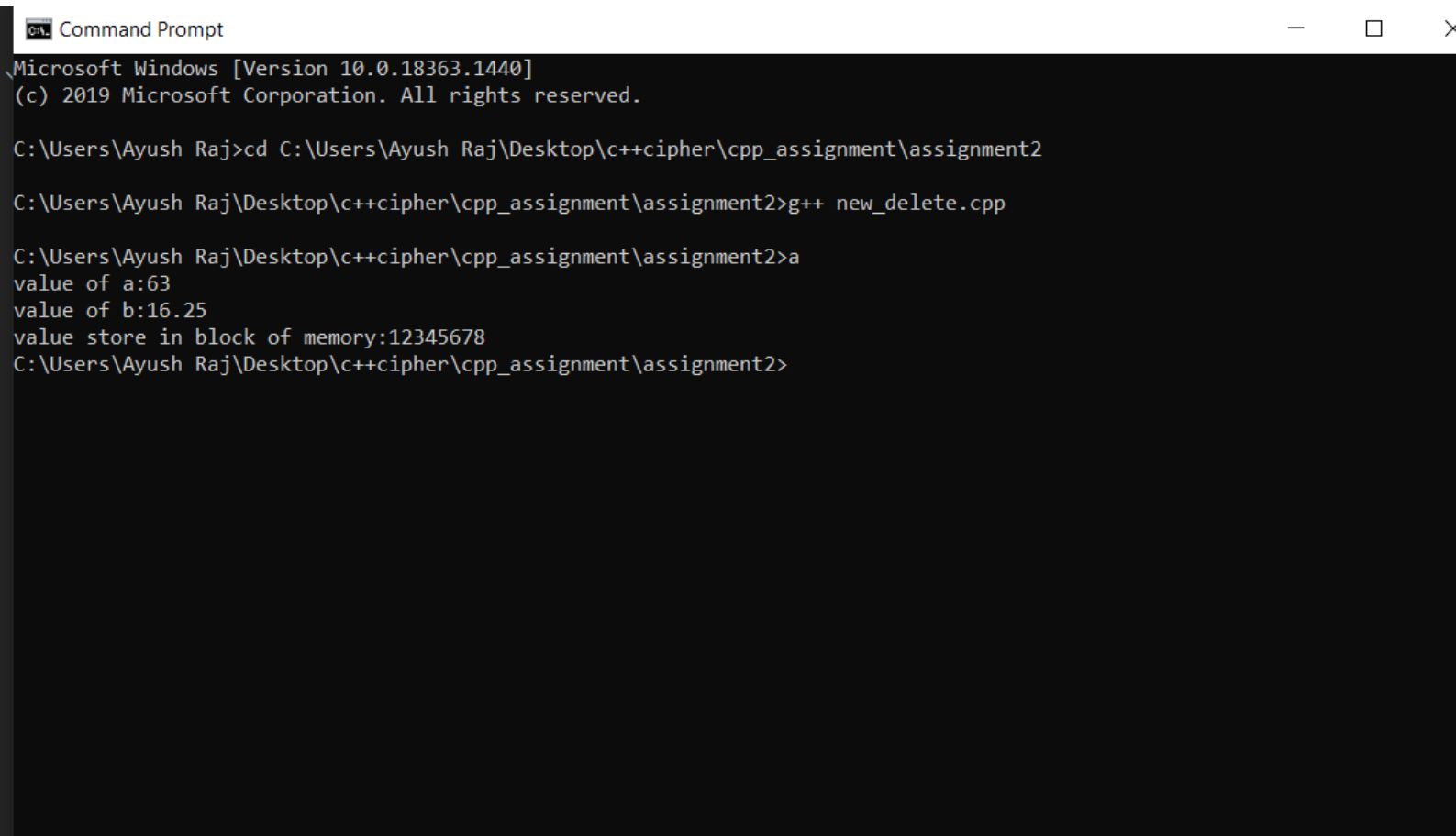
Ans-> Delete Keyword is used to deallocate the memory. Delete is an operator that is used to destroy array and non-array (pointer) objects which are created by new expression.

The new operator denotes a request for memory allocation on the Free Store. If sufficient memory is available, new operator initializes the memory and returns the address of the newly allocated and initialized memory to the pointer variable.

Q.2-> What are constructors? Why they are required? Explain different types of Constructors with Suitable example.

Ans-> Constructor in C++ is a special method that is automatically


File Edit Selection View Go Run Terminal Help
new_delete.cpp - assignment2 - Visual Studio Code
EXPLORER
ASSIGNMENT2
a.exe
fnoverloading.cpp
member.cpp
new_delete.cpp
sortarray.cpp
Welcome
new_delete.cpp
new_delete.cpp > main()
#include<bits/stdc++.h>
using namespace std;
int main()
{
int*a=NULL;
a=new(nothrow)int;
if(!a)
cout<<"allocation of memory failed\n";
else
{
*a=89;
cout<<"value of a:"<<*a<<endl;
}
float *b=new float(12.86);
cout<<"value of b:"<<*b<<endl;
int n=8;
int *c=new(nothrow) int[n];
if (!c)
cout<<"allocation of memory failed\n";
else
{
for(int i=0;i<n;i++)
b[i]=i+1;
cout<<"value store in block of memory:";
for(int i=0;i<n;i++)
cout<<b[i]<<"";
}
delete a;
delete c;
delete[]b;
return 0;
}
OUTLINE


Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Ayush Raj>cd C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>g++ new_delete.cpp
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>a
value of a:63
value of b:16.25
value store in block of memory:12345678
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>

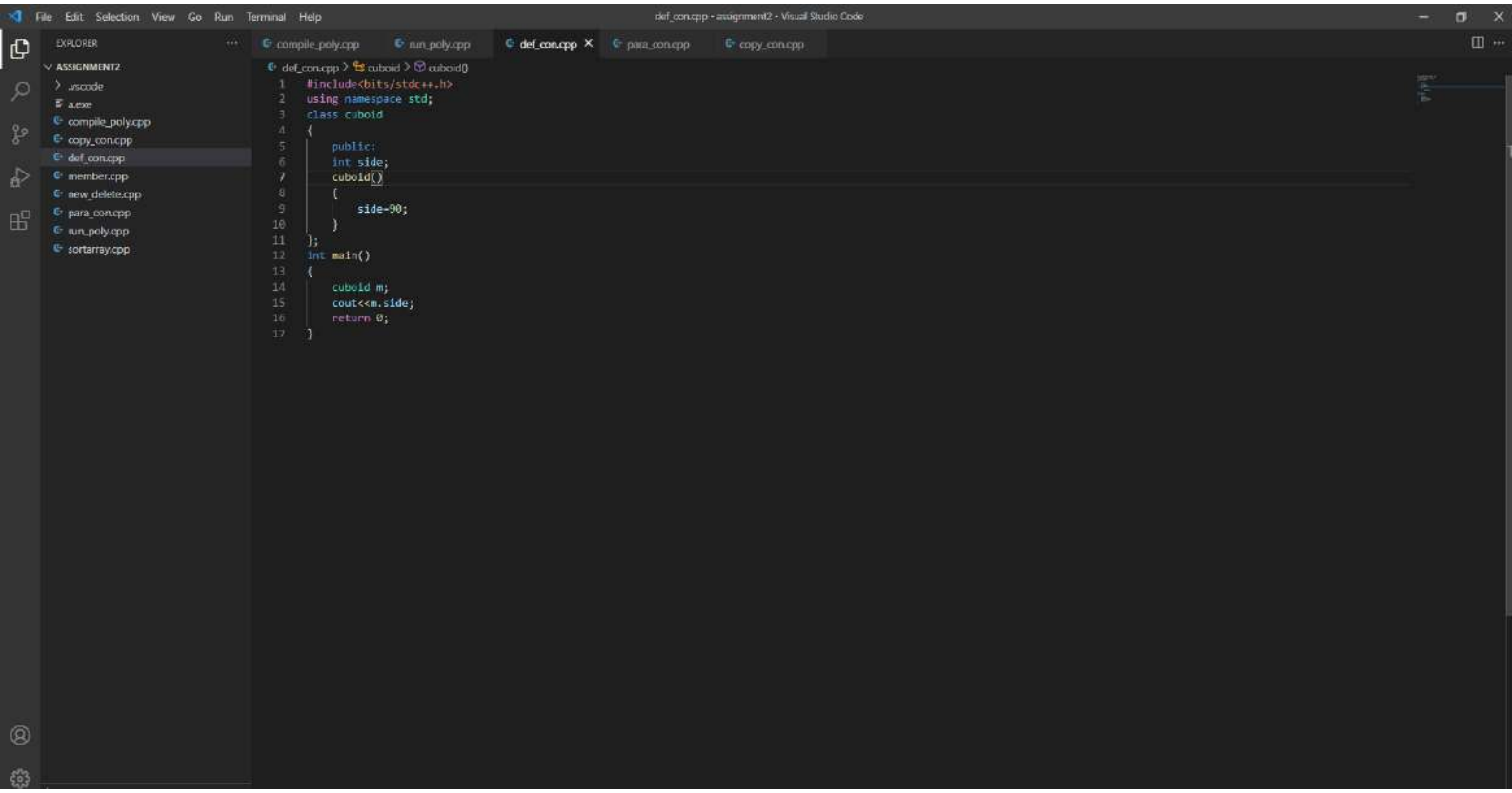
Called when an object of class is created.
The main purpose of the class Constructor in c++ is to construct an object of the class. It is used to initialize all class data members.

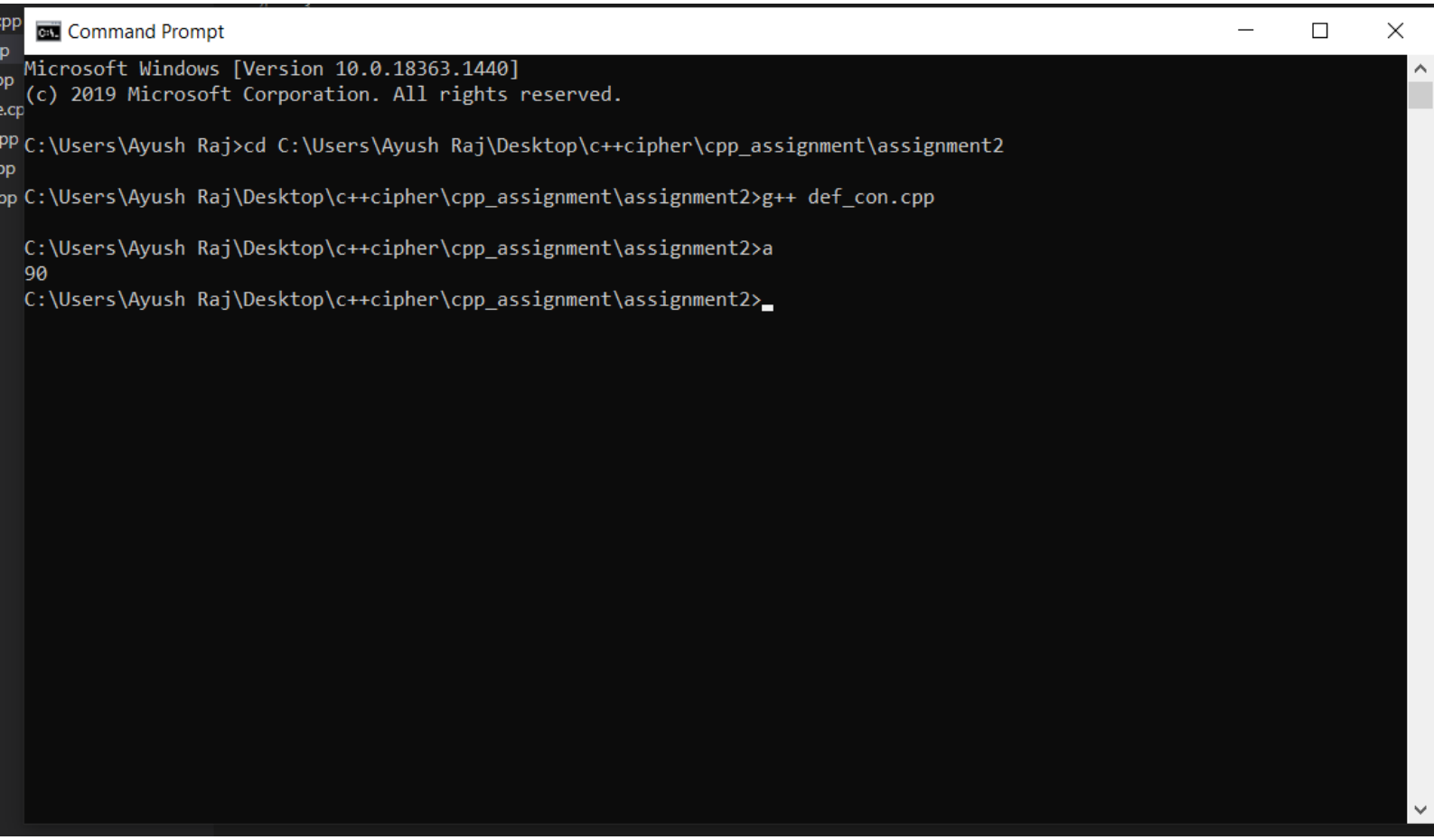
Types of Constructors:-

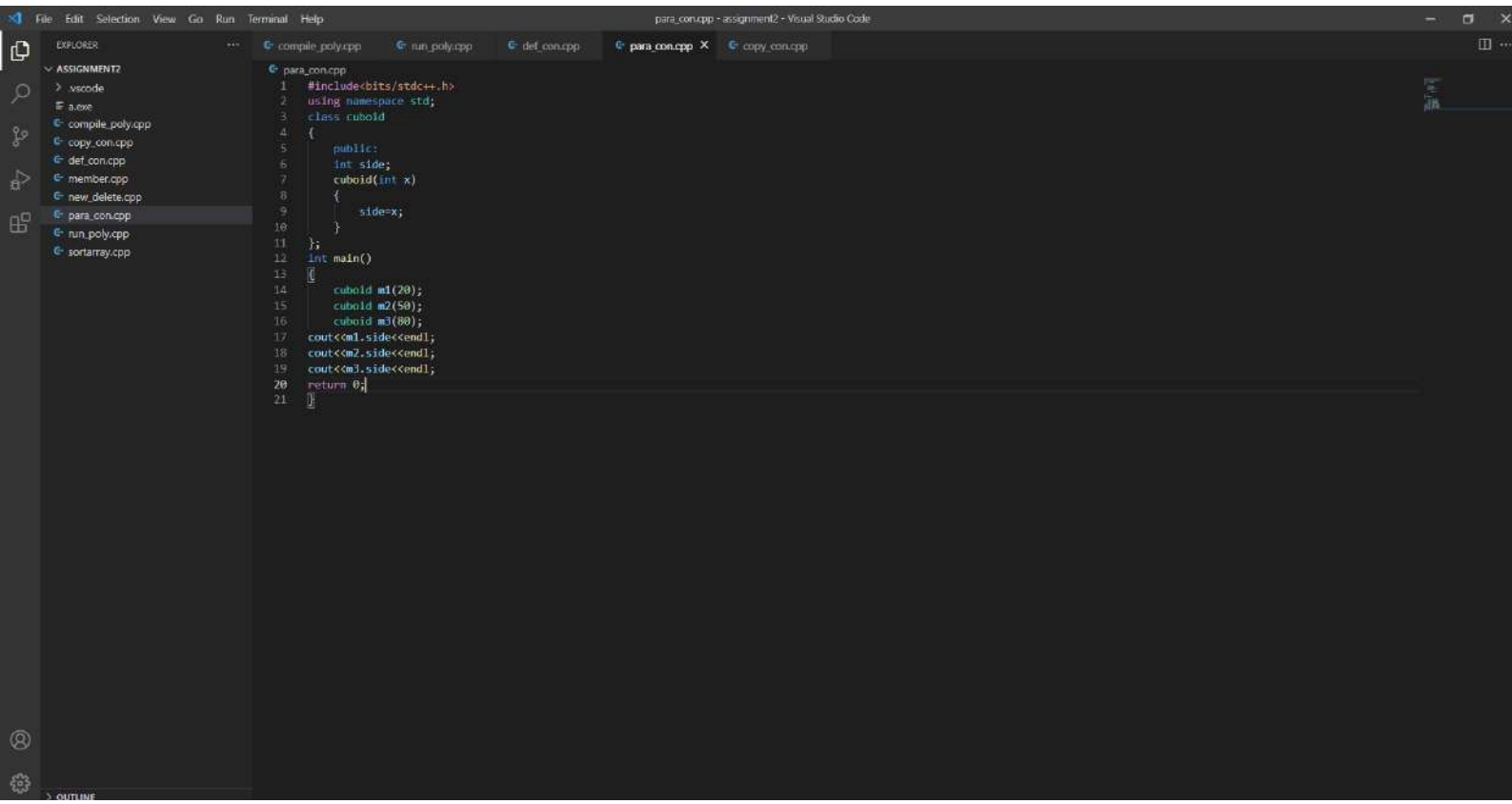
1. Default Constructor:- It does not take any argument. It has no parameter.

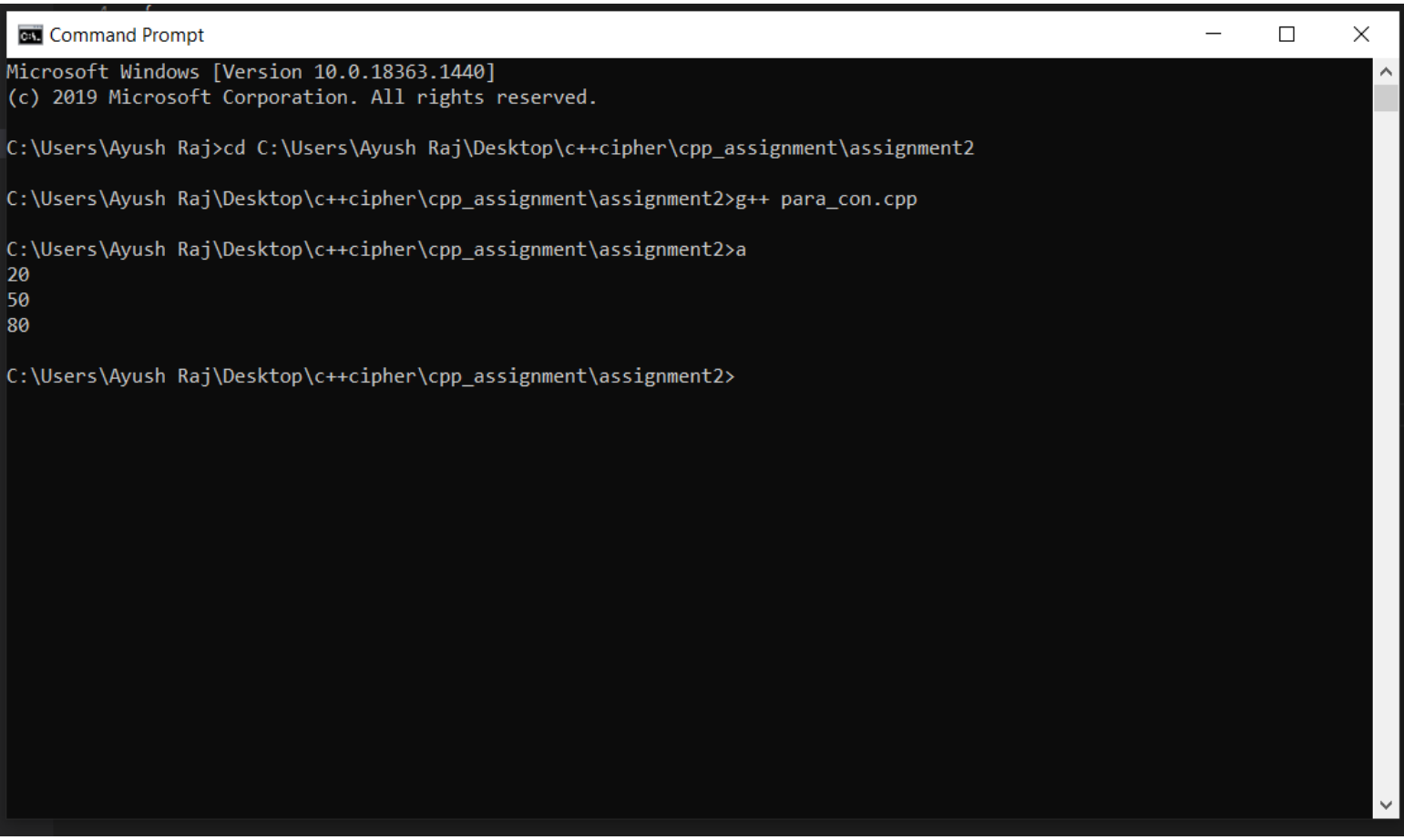
2. Parameterized Constructors:- It has Parameter. We can provide different values to data members of different objects, by passing the appropriate values as argument.

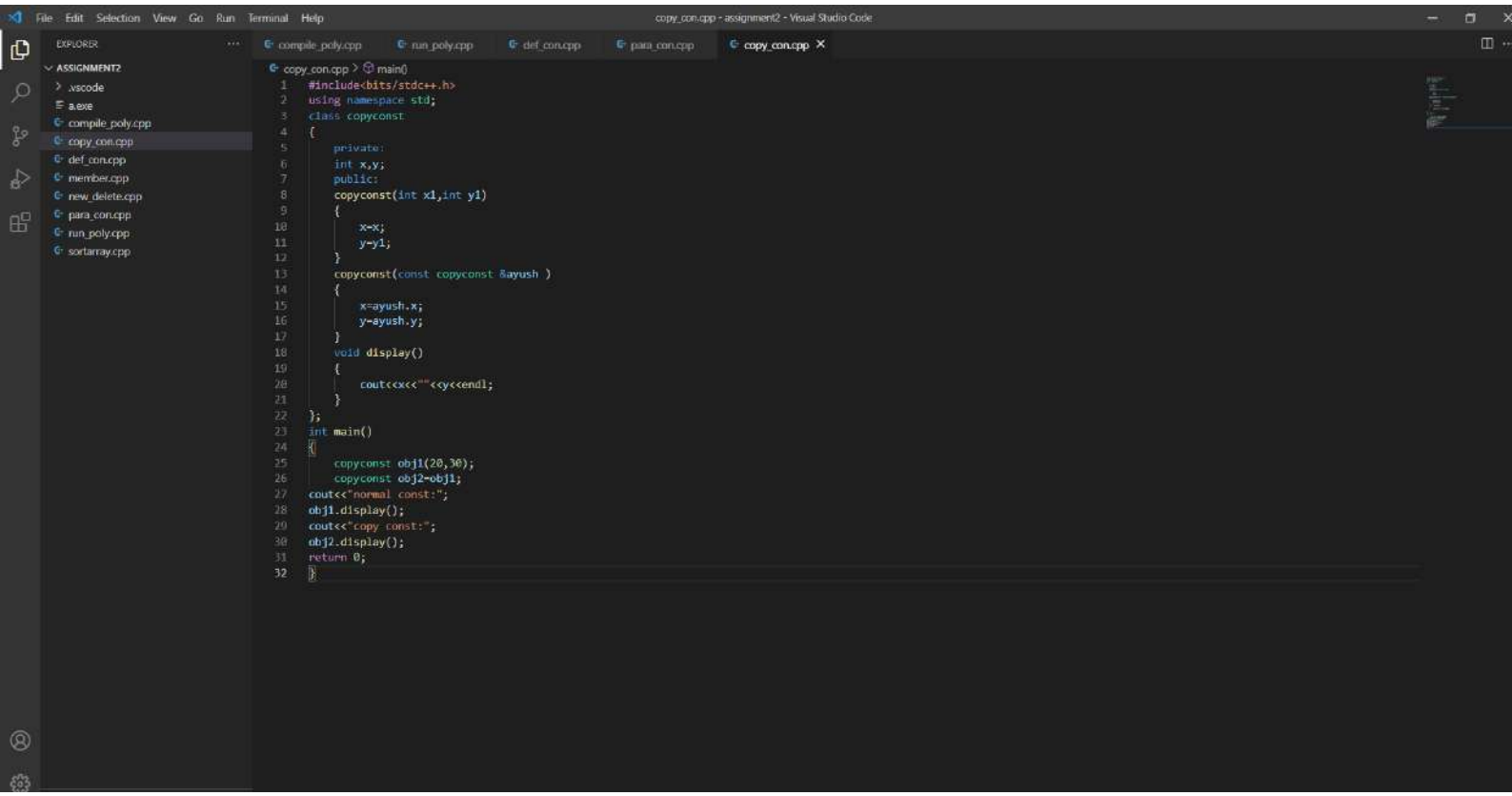
3) Copy Constructor:- It is used to create a copy of an already existing object of a class type. It is usually of the form X(x&), where X is the class name. The Compiler provides a default Copy Constructor to all the Classes.

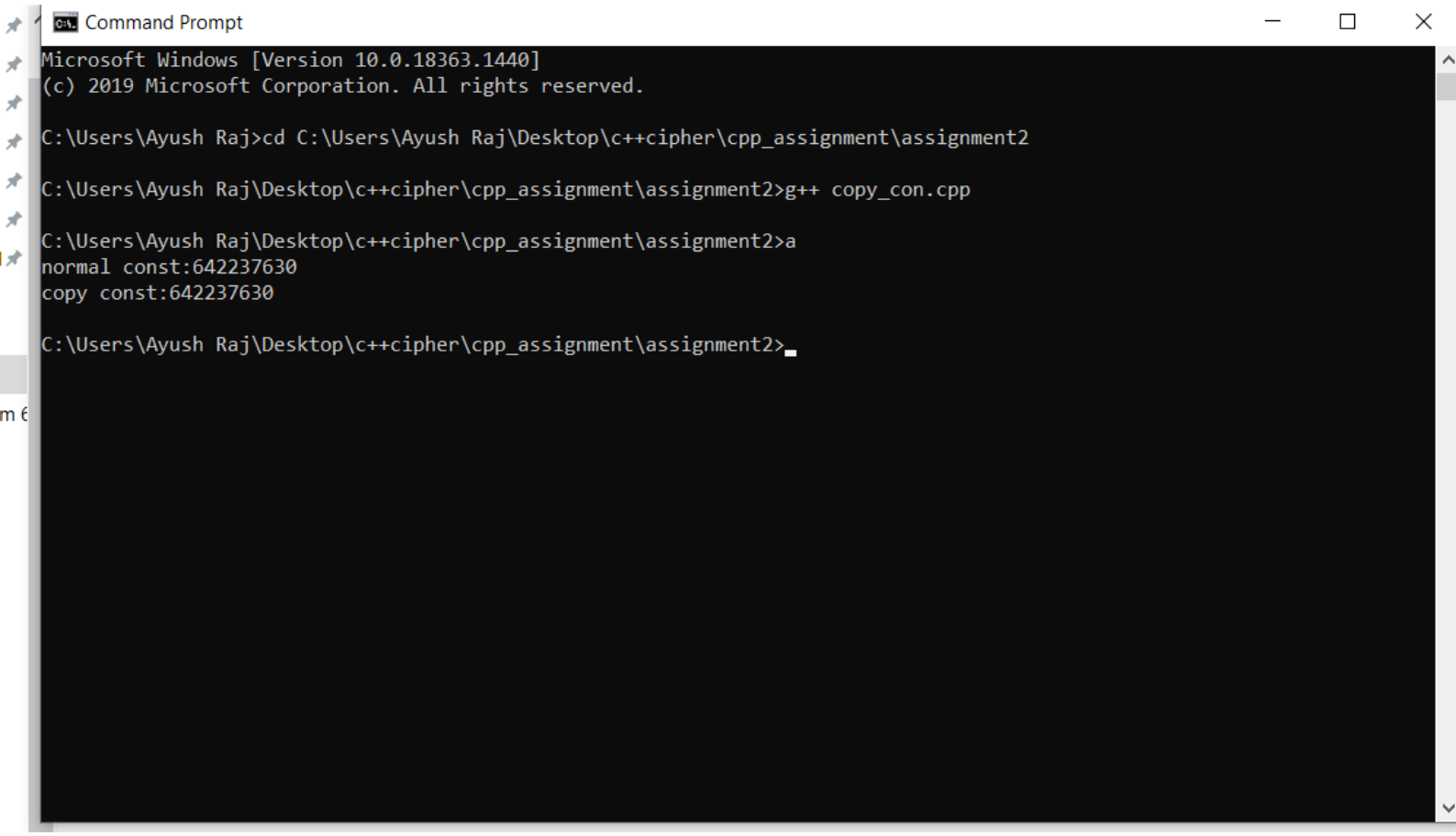

File Edit Selection View Go Run Terminal Help
def_con.cpp - assignment2 - Visual Studio Code
EXPLORER
ASSIGNMENT2
.vscode
a.exe
compile_poly.cpp
copy_con.cpp
def_con.cpp
member.cpp
new_delete.cpp
para_con.cpp
run_poly.cpp
sortarray.cpp
compile_poly.cpp
run_poly.cpp
def_con.cpp
para_con.cpp
copy_con.cpp
def_con.cpp > cuboid > cuboid()
#include<bits/stdc++.h>
using namespace std;
class cuboid
{
public:
int side;
cuboid()
{
side=90;
}
};
int main()
{
cuboid m;
cout<<m.side;
return 0;
}


Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Ayush Raj>cd C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>g++ def_con.cpp
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>a
90
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>


File Edit Selection View Go Run Terminal Help
para_con.cpp - assignment2 - Visual Studio Code
EXPLORER
ASSIGNMENT2
.vscode
a.exe
compile_poly.cpp
copy_con.cpp
def_con.cpp
member.cpp
new_delete.cpp
para_con.cpp
run_poly.cpp
sortarray.cpp
OUTLINE
compile_poly.cpp
run_poly.cpp
def_con.cpp
para_con.cpp
copy_con.cpp
para_con.cpp
#include<bits/stdc++.h>
using namespace std;
class cuboid
{
public:
int side;
cuboid(int x)
{
side=x;
}
};
int main()
{
cuboid m1(20);
cuboid m2(50);
cuboid m3(80);
cout<<m1.side<<endl;
cout<<m2.side<<endl;
cout<<m3.side<<endl;
return 0;
}


Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Ayush Raj>cd C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>g++ para_con.cpp
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>a
20
50
80
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>


File Edit Selection View Go Run Terminal Help
copy_con.cpp - assignment2 - Visual Studio Code
EXPLORER
ASSIGNMENT2
.vscode
a.exe
compile_poly.cpp
copy_con.cpp
def_con.cpp
member.cpp
new_delete.cpp
para_con.cpp
run_poly.cpp
sortarray.cpp
compile_poly.cpp
run_poly.cpp
def_con.cpp
para_con.cpp
copy_con.cpp
copy_con.cpp > main()
#include<bits/stdc++.h>
using namespace std;
class copyconst
{
private:
int x,y;
public:
copyconst(int x1,int y1)
{
x=x;
y=y1;
}
copyconst(const copyconst &ayush)
{
x=ayush.x;
y=ayush.y;
}
void display()
{
cout<<x<<""<<y<<endl;
}
};
int main()
{
copyconst obj1(20,30);
copyconst obj2=obj1;
cout<<"normal const:";
obj1.display();
cout<<"copy const:";
obj2.display();
return 0;
}


Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Ayush Raj>cd C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>g++ copy_con.cpp
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>a
normal const:642237630
copy const:642237630
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>

Q.3> Explain the difference b/w object oriented and procedural programming language in detail.

Ans->

Object-Oriented	Procedural
i) In OOP, program is divided into small parts called objects	i) In procedural programming language, program is divided into small parts called functions.
ii) It follow bottom up approach.	ii) It follows top-down approach
iii) It can access specifier like public, private and protected etc.	iii) There is no access specifier in procedural programming.
iv) Adding new data and function is easy.	iv) Adding new data and function is not easy.

v) Object oriented programming provides data hiding so it is more secure.	v) It does not have any proper way for hiding data so it is less secure.
vi) Overloading is possible in oop.	vi) In this overloading is not possible.
vii) In oops data is more important than function.	vii) In procedural function is more important than data.
viii) It is based on real world.	viii) It is based on unreal world.
ix) Examples:- C++, Java, Python, C# etc	ix) Example:- C, FORTAN, Pascal, Basic, etc.

2^{n-1}

Page No.: 07
Date: / /

Long answer Type question:

A) Explain the type of polymorphism with code.

Ans→ Polymorphism means many form.
Two types of polymorphism:-
(i) Compile time.
(ii) Run time

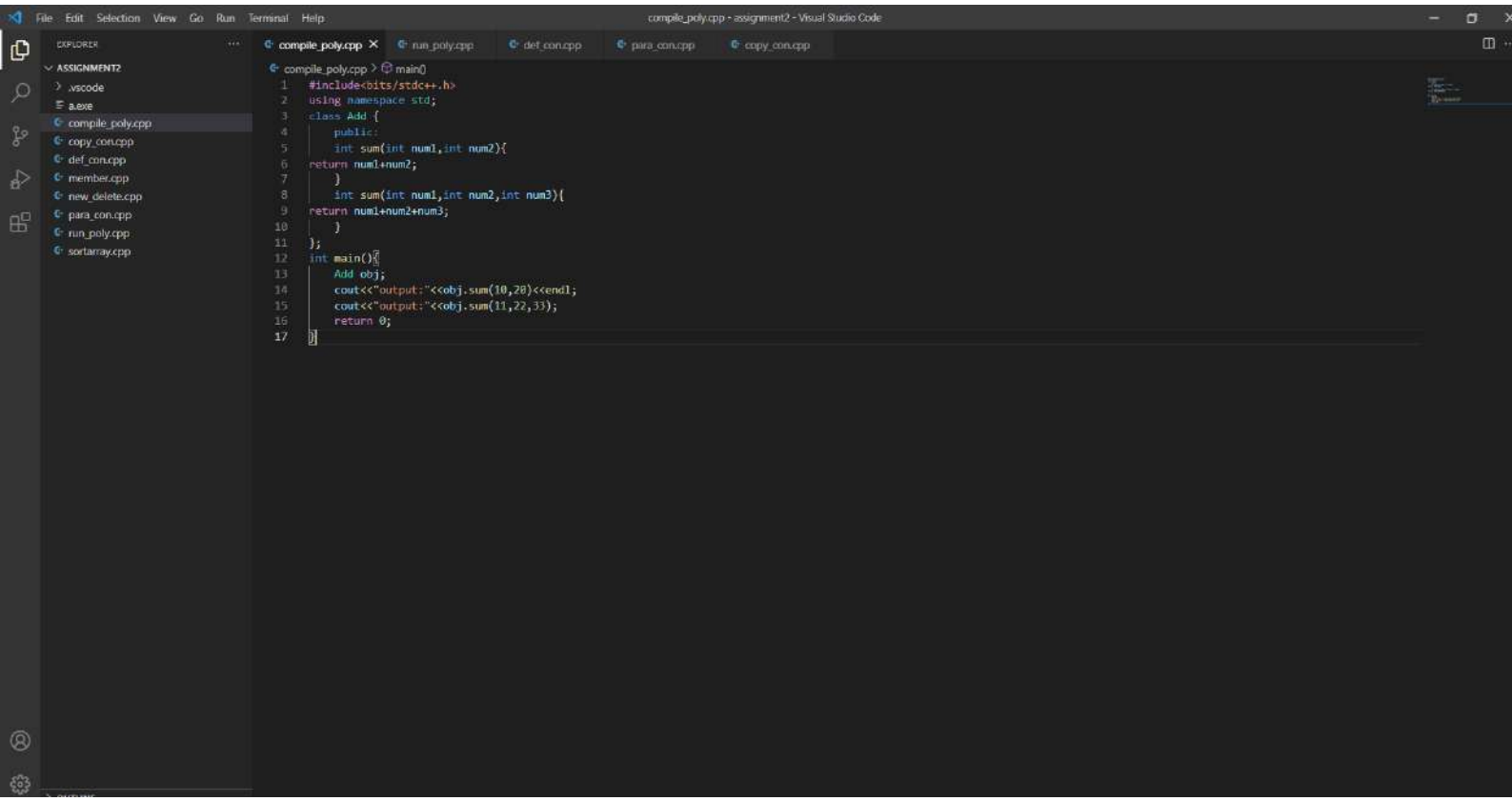
(i) Compile time polymorphism:-
This is also known as Static (or early) binding. Function overloading and operator overloading are perfect example of compile time polymorphism.
Example:- In this example we have two functions with same name but differen number of arguments. Based on how many parameters 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.

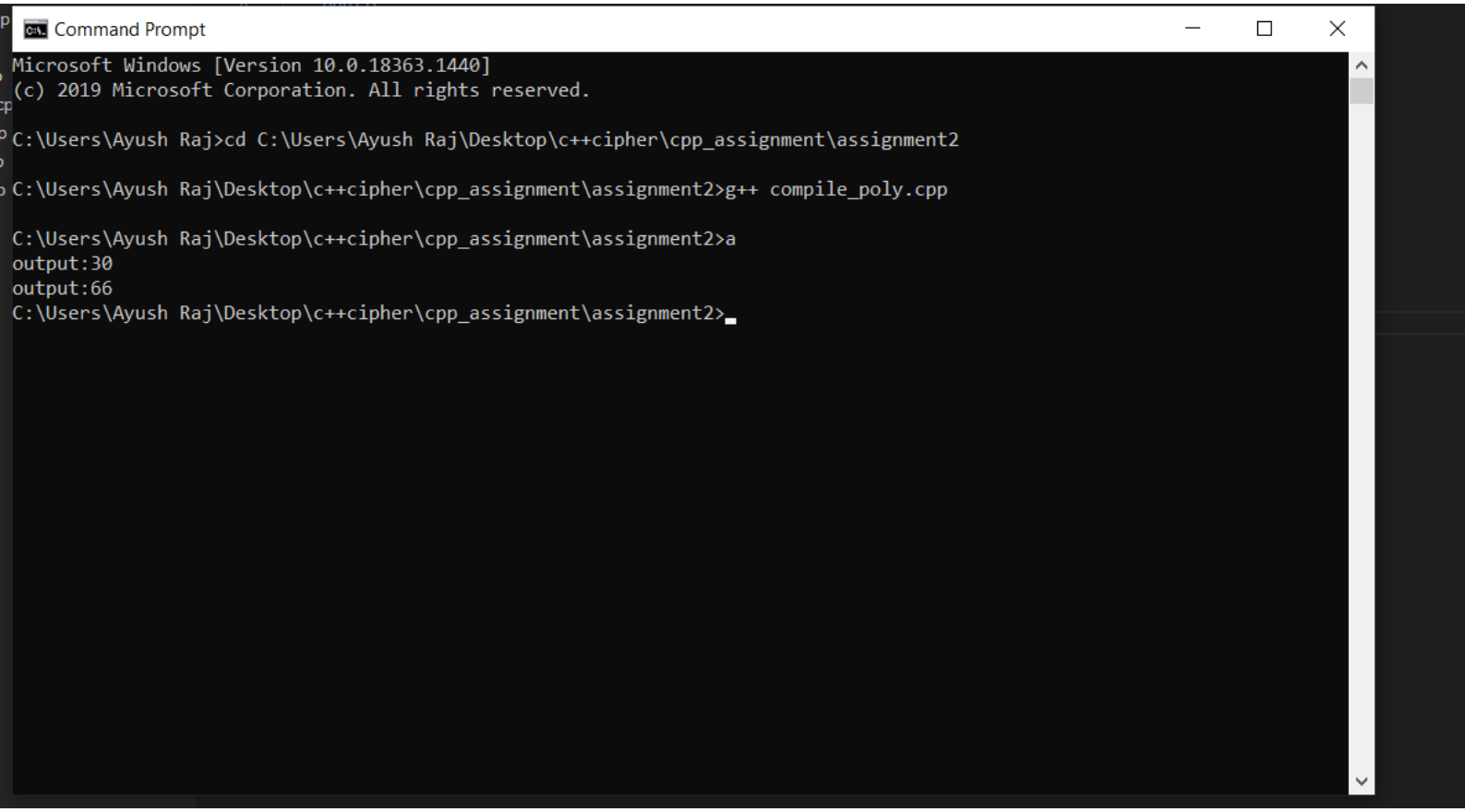
ii) Runtime polymorphism:- This is also known as dynamic (or late) binding.

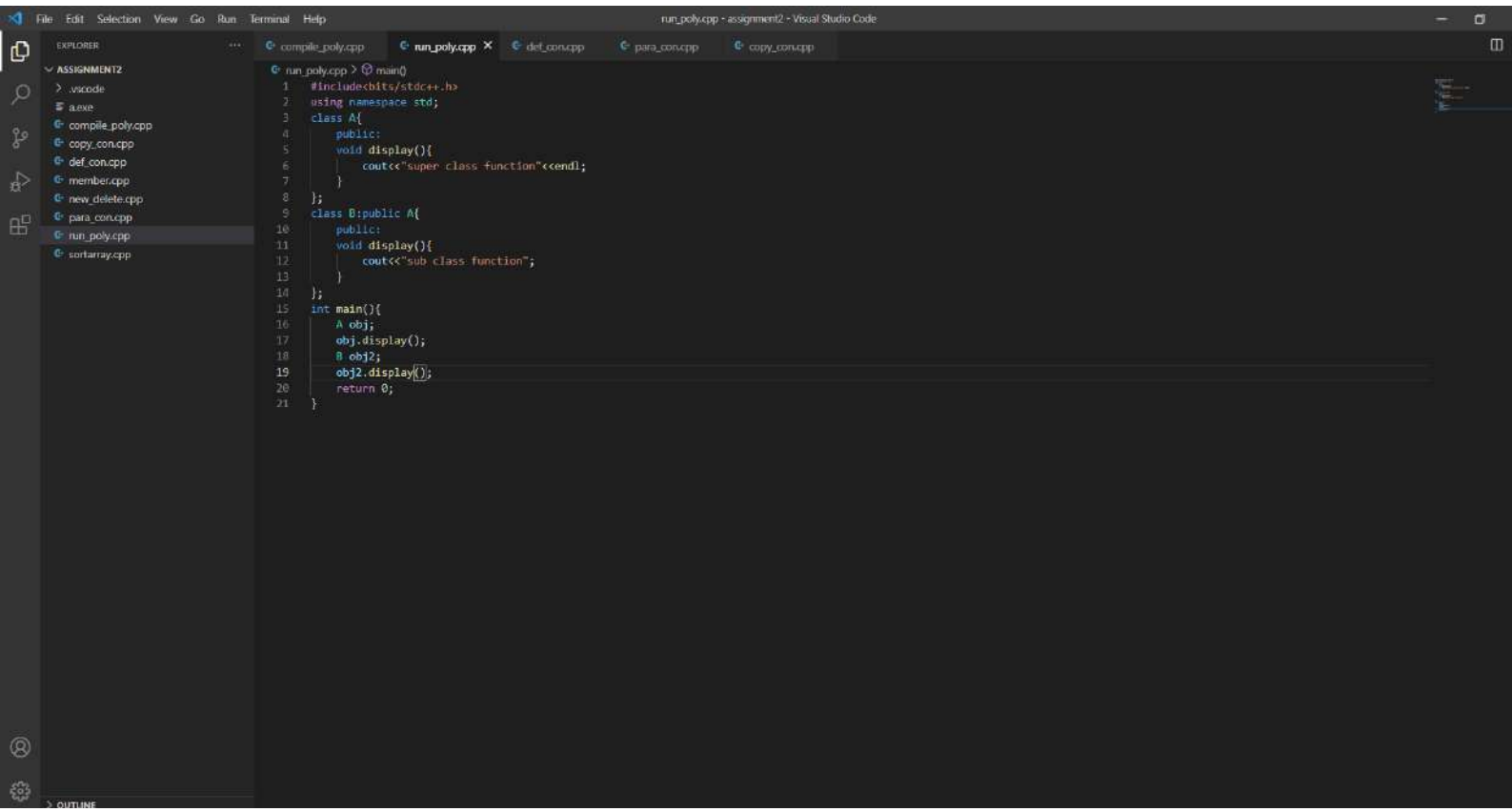
Example:- Function overriding is an example of runtime polymorphism.

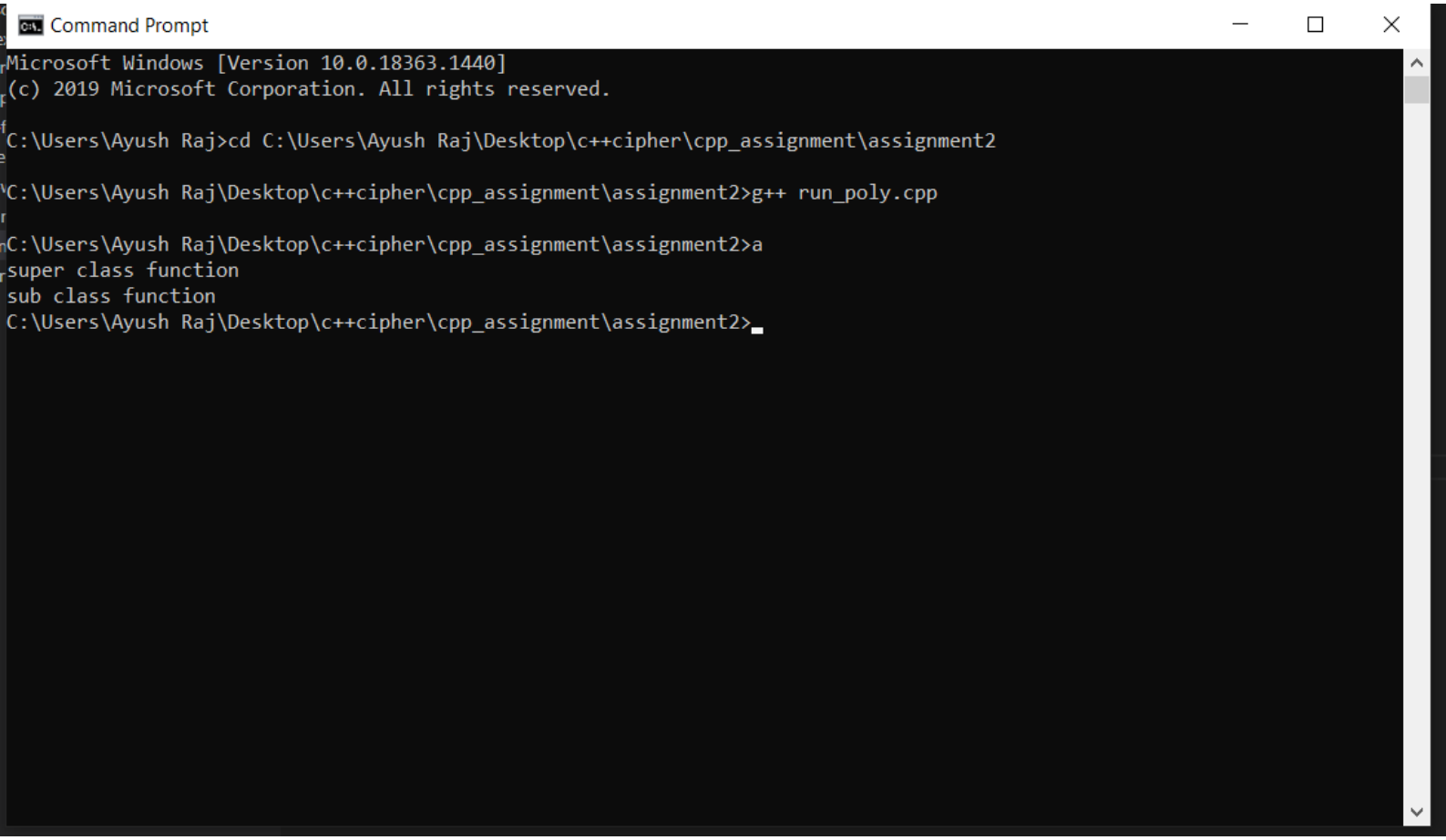
Function Overriding:- When child class declares a method which is already present in the parent class then this is called function overriding, here Child class overrides the parent class.

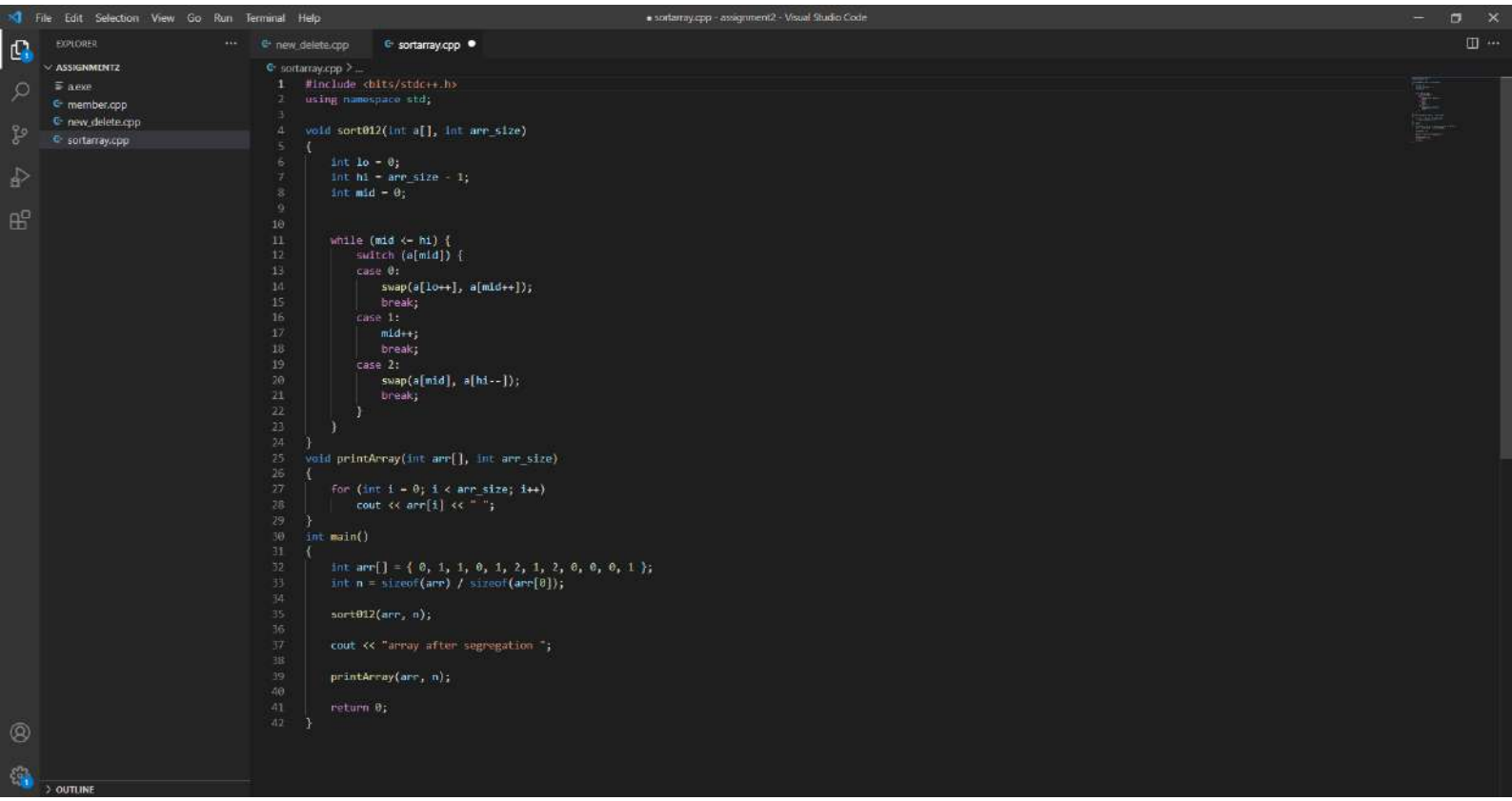
In case of function overriding we have two ~~diffe~~ definitions of same function. One is parent class and one is Child class. The call to the function is determined at runtime to decide which definition of the function is to be called, that is the reason it is called runtime polymorphism.

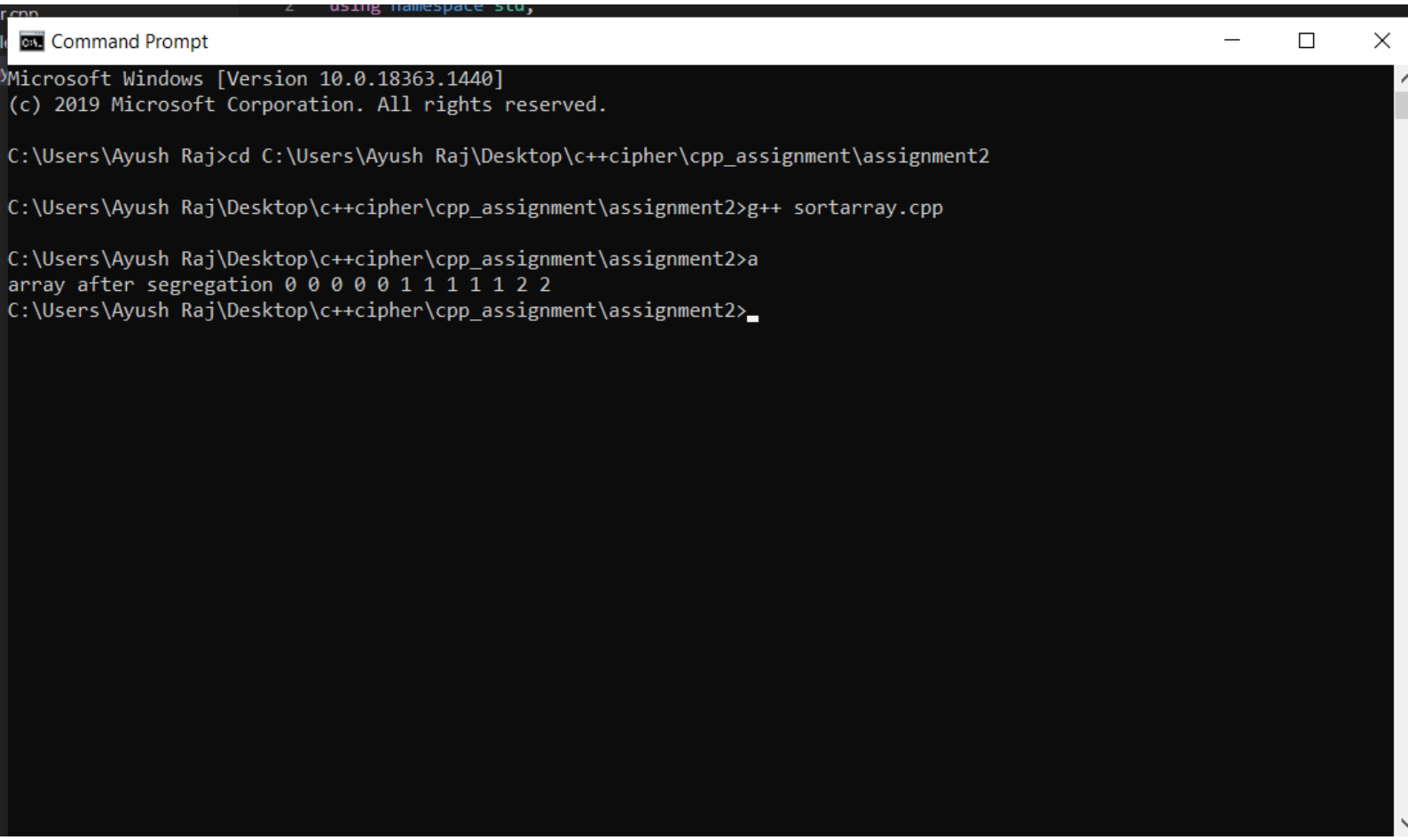

File Edit Selection View Go Run Terminal Help
compile_poly.cpp - assignment2 - Visual Studio Code
EXPLORER
ASSIGNMENT2
.vscode
a.exe
compile_poly.cpp
copy_con.cpp
def_con.cpp
member.cpp
new_delete.cpp
para_con.cpp
run_poly.cpp
sortarray.cpp
compile_poly.cpp
run_poly.cpp
def_con.cpp
para_con.cpp
copy_con.cpp
compile_poly.cpp > main()
#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;
}

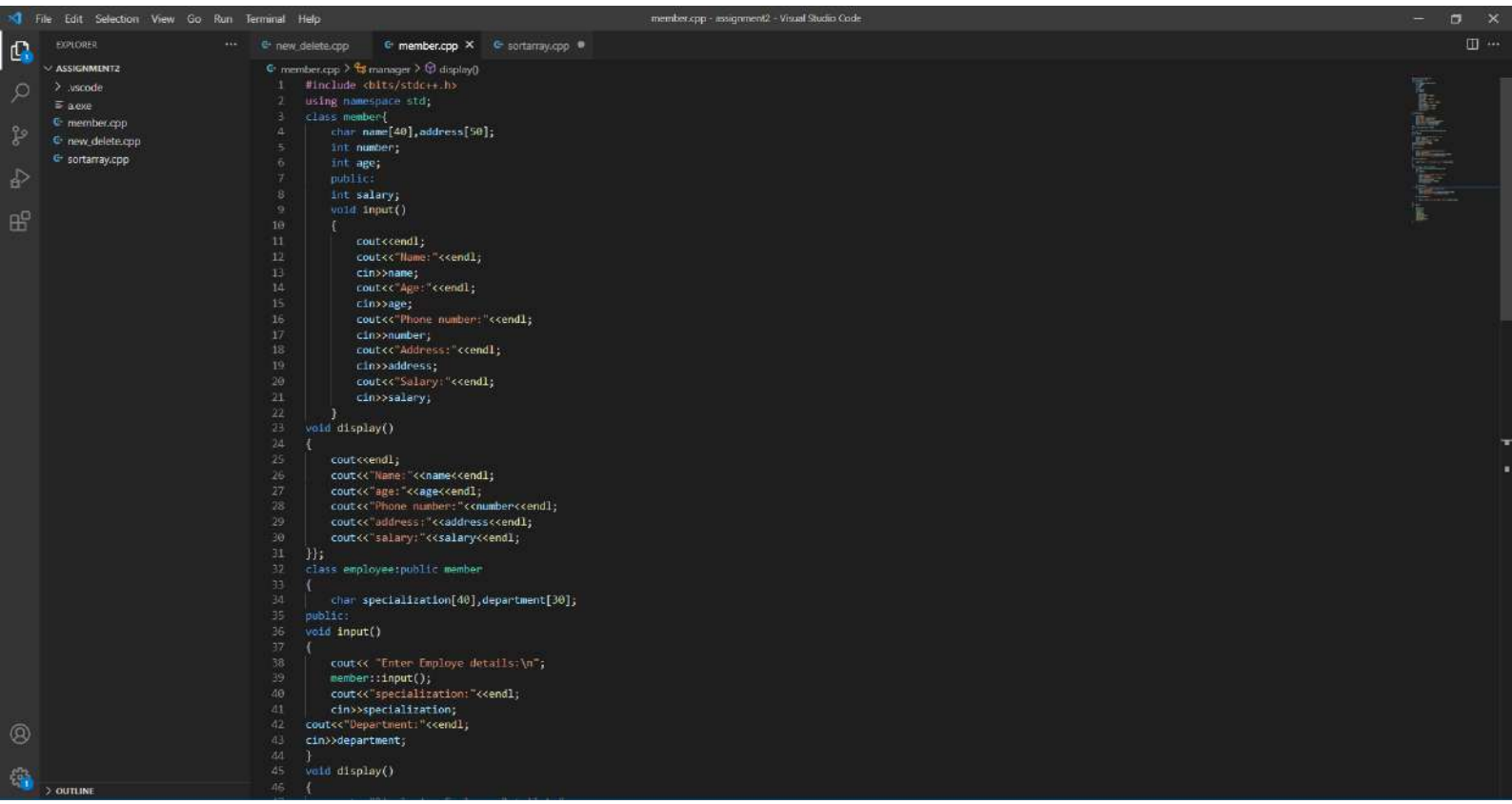

Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Ayush Raj>cd C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>g++ compile_poly.cpp
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>a
output:30
output:66
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>

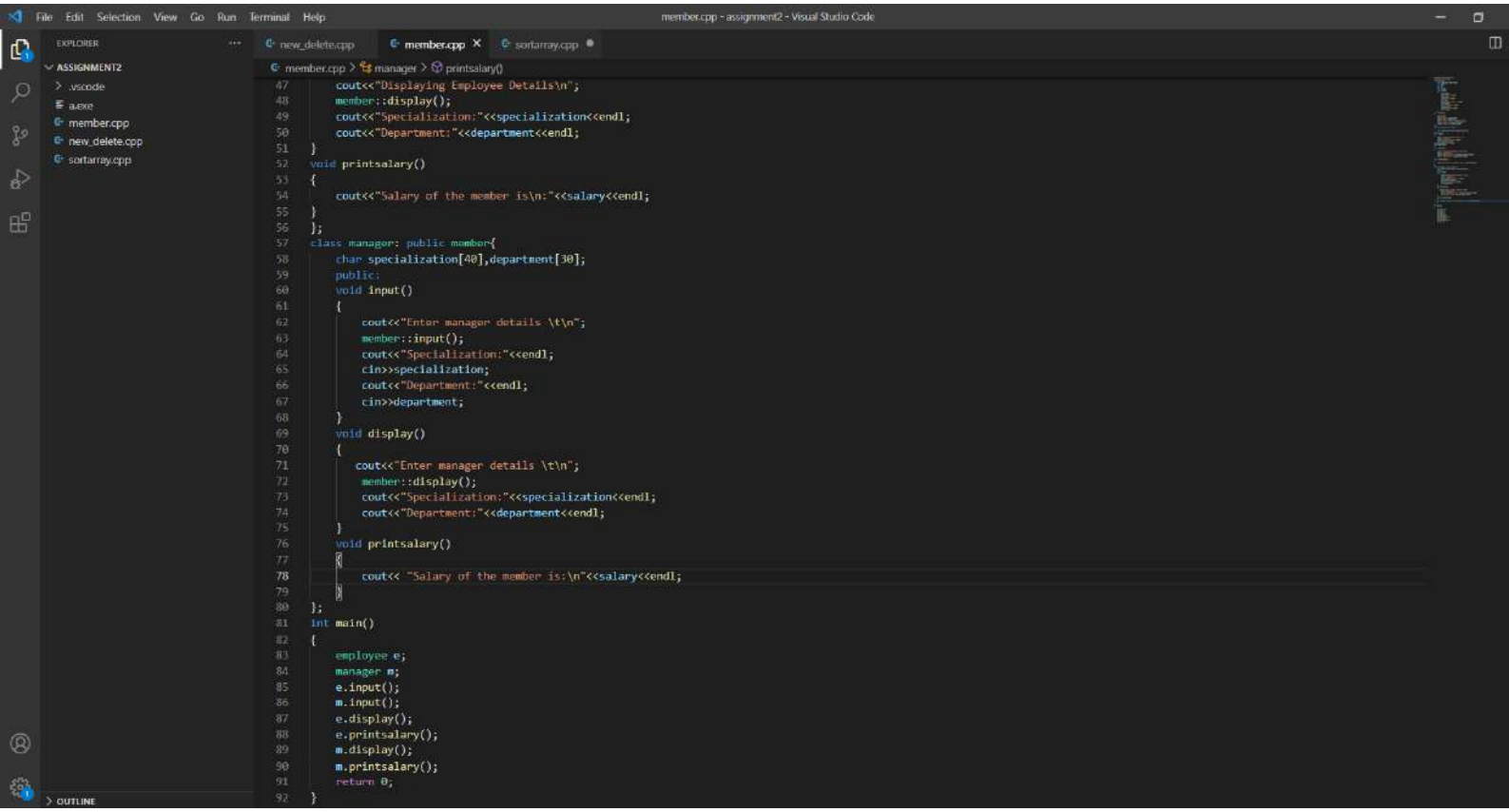

File Edit Selection View Go Run Terminal Help
run_poly.cpp - assignment2 - Visual Studio Code
EXPLORER
ASSIGNMENT2
.vscode
a.exe
compile_poly.cpp
copy_con.cpp
def_con.cpp
member.cpp
new_delete.cpp
para_con.cpp
run_poly.cpp
sortarray.cpp
OUTLINE
compile_poly.cpp
run_poly.cpp
def_con.cpp
para_con.cpp
copy_con.cpp
run_poly.cpp > main()
#include<bits/stdc++.h>
using namespace std;
class A{
public:
void display(){
cout<<"super class function"<<endl;
}
};
class B:public A{
public:
void display(){
cout<<"sub class function";
}
};
int main(){
A obj;
obj.display();
B obj2;
obj2.display();
return 0;
}


Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Ayush Raj>cd C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>g++ run_poly.cpp
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>a
super class function
sub class function
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>

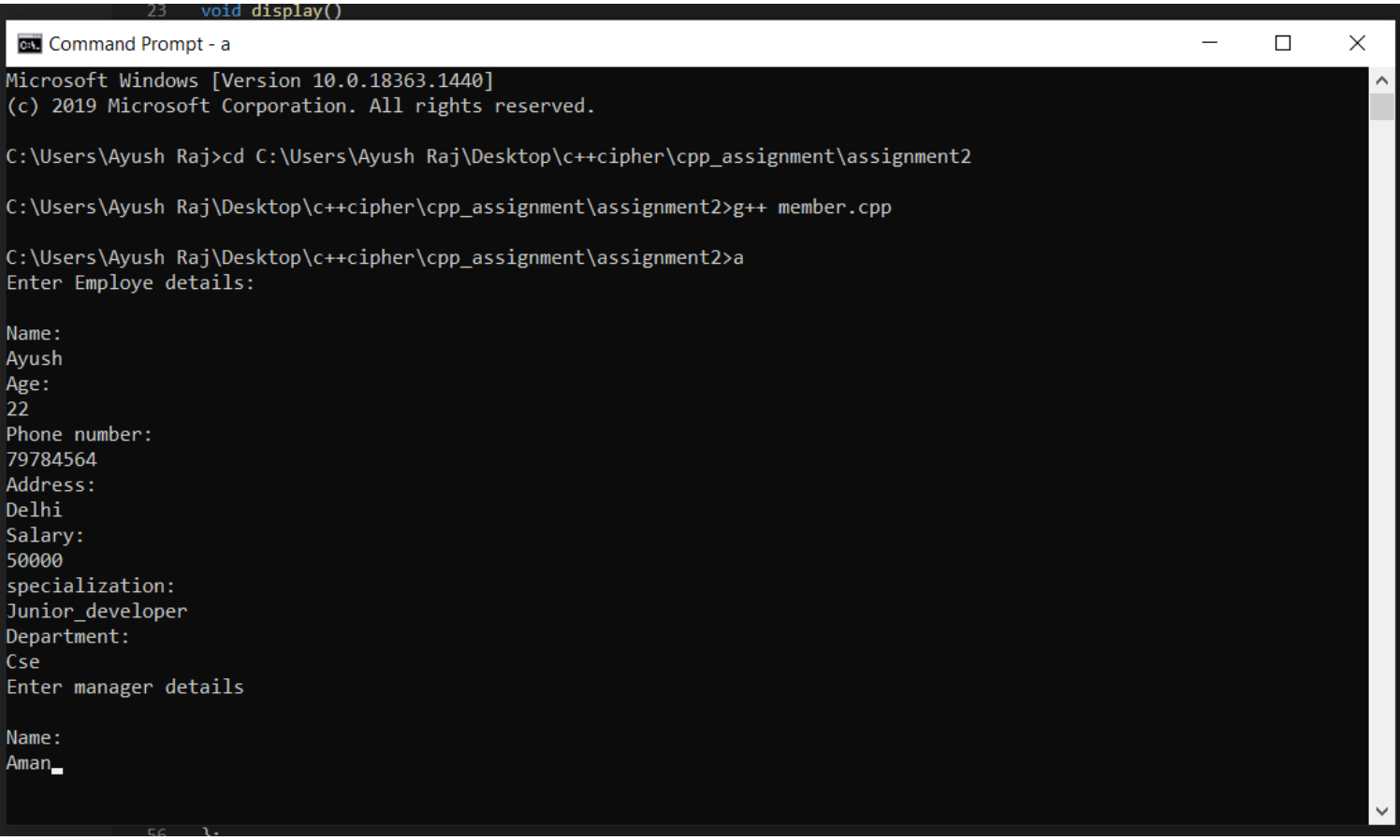

File Edit Selection View Go Run Terminal Help
• sortarray.cpp - assignment2 - Visual Studio Code
EXPLORER
ASSIGNMENT2
a.exe
member.cpp
new_delete.cpp
sortarray.cpp
new_delete.cpp
sortarray.cpp •
sortarray.cpp > ...
#include <bits/stdc++.h>
using namespace std;
void sort012(int a[], int arr_size)
{
int 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 printArray(int arr[], int arr_size)
{
for (int i = 0; i < arr_size; i++)
cout << arr[i] << " ";
}
int main()
{
int arr[] = { 0, 1, 1, 0, 1, 2, 1, 2, 0, 0, 0, 1 };
int n = sizeof(arr) / sizeof(arr[0]);
sort012(arr, n);
cout << "array after segregation ";
printArray(arr, n);
return 0;
}
OUTLINE


Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Ayush Raj>cd C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>g++ sortarray.cpp
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>a
array after segregation 0 0 0 0 0 1 1 1 1 1 2 2
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>


File Edit Selection View Go Run Terminal Help
member.cpp - assignment2 - Visual Studio Code
EXPLORER
ASSIGNMENT2
.vscode
a.exe
member.cpp
new_delete.cpp
sortarray.cpp
new_delete.cpp
member.cpp
sortarray.cpp
member.cpp > manager > display()
#include <bits/stdc++.h>
using namespace std;
class member{
char name[40],address[50];
int number;
int age;
public:
int salary;
void input()
{
cout<<endl;
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;
}
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[40],department[30];
public:
void input()
{
cout<< "Enter Employe details:\n";
member::input();
cout<<"specialization:"<<endl;
cin>>specialization;
cout<<"Department:"<<endl;
cin>>department;
}
void display()
{
OUTLINE

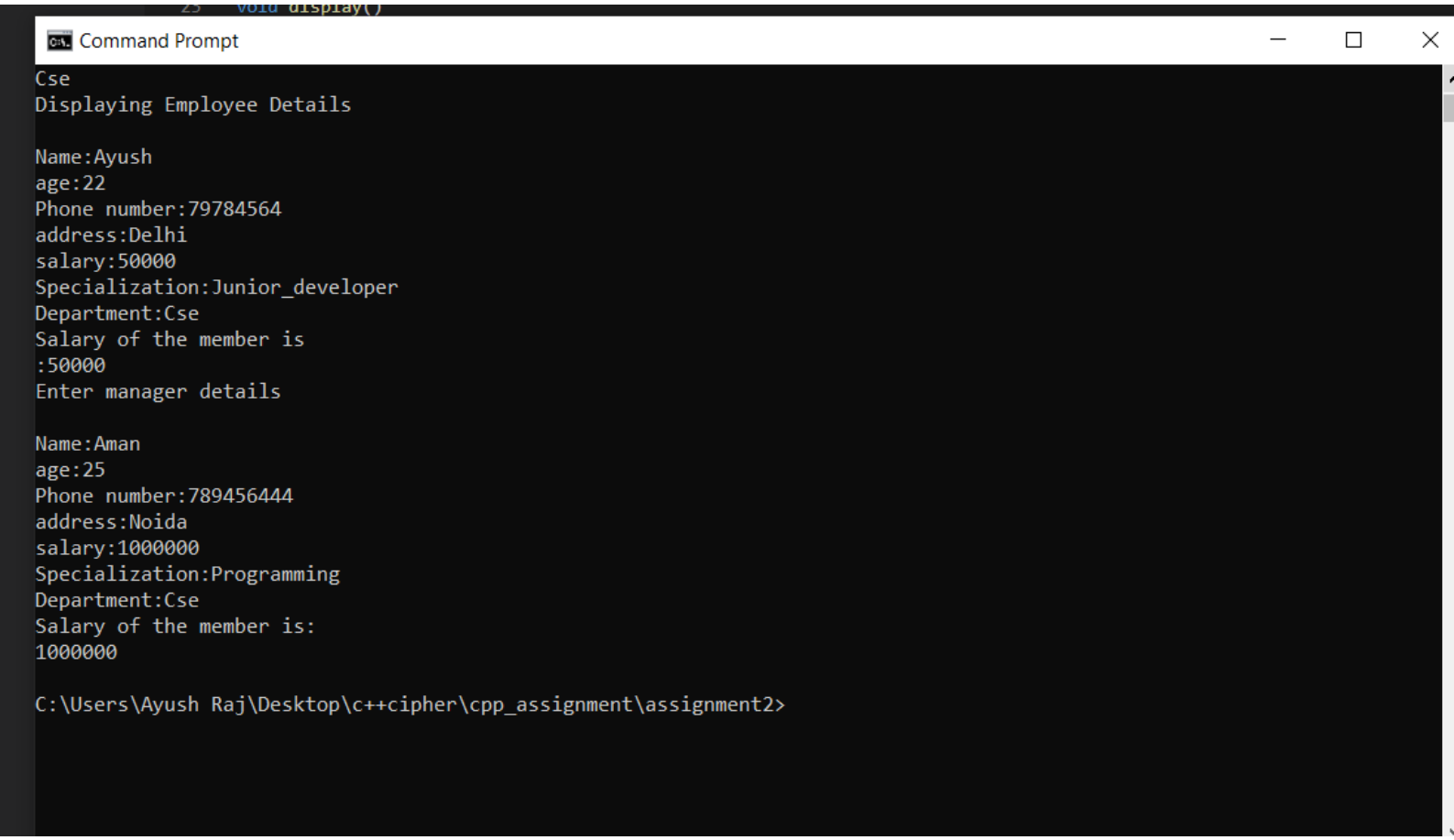

File Edit Selection View Go Run Terminal Help
member.cpp - assignment2 - Visual Studio Code
EXPLORER
ASSIGNMENT2
.vscode
a.exe
member.cpp
new_delete.cpp
sortarray.cpp
new_delete.cpp
member.cpp
sortarray.cpp
member.cpp > manager > printsalary()
cout<<"Displaying Employee Details\n";
member::display();
cout<<"Specialization:"<<specialization<<endl;
cout<<"Department:"<<department<<endl;
}
void printsalary()
{
cout<<"Salary of the member is\n:"<<salary<<endl;
}
};
class manager: public member{
char specialization[40],department[30];
public:
void input()
{
cout<<"Enter manager details \t\n";
member::input();
cout<<"Specialization:"<<endl;
cin>>specialization;
cout<<"Department:"<<endl;
cin>>department;
}
void display()
{
cout<<"Enter manager details \t\n";
member::display();
cout<<"Specialization:"<<specialization<<endl;
cout<<"Department:"<<department<<endl;
}
void printsalary()
{
cout<< "Salary of the member is:\n"<<salary<<endl;
}
};
int main()
{
employee e;
manager m;
e.input();
m.input();
e.display();
e.printsalary();
m.display();
m.printsalary();
return 0;
}
OUTLINE

```
23    void display()
```




```
56    };
```


Command Prompt
Delhi
Salary:
50000
specialization:
Junior_developer
Department:
Cse
Enter manager details
Name:
Aman
Age:
25
Phone number:
789456444
Address:
Noida
Salary:
1000000
Specialization:
Programming
Department:
Cse
Displaying Employee Details
Name:Ayush
age:22
Phone number:79784564
address:Delhi
salary:50000


Command Prompt
Cse
Displaying Employee Details
Name:Ayush
age:22
Phone number:79784564
address:Delhi
salary:50000
Specialization:Junior_developer
Department:Cse
Salary of the member is
:50000
Enter manager details
Name:Aman
age:25
Phone number:789456444
address:Noida
salary:1000000
Specialization:Programming
Department:Cse
Salary of the member is:
1000000
C:\Users\Ayush Raj\Desktop\c++cipher\cpp_assignment\assignment2>