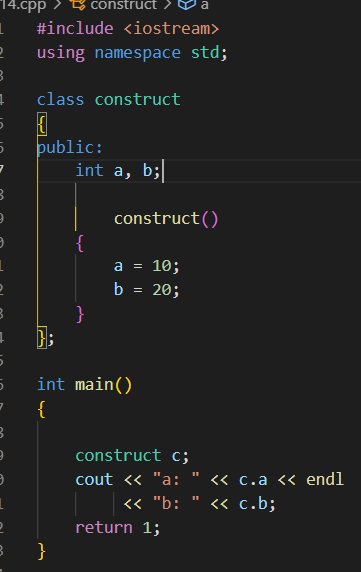
**Lab – 4**

Constructor, destructors, dynamic memory management, friend function

1. Write a program in C++ to implement various constructors

(a) No argument

Code:

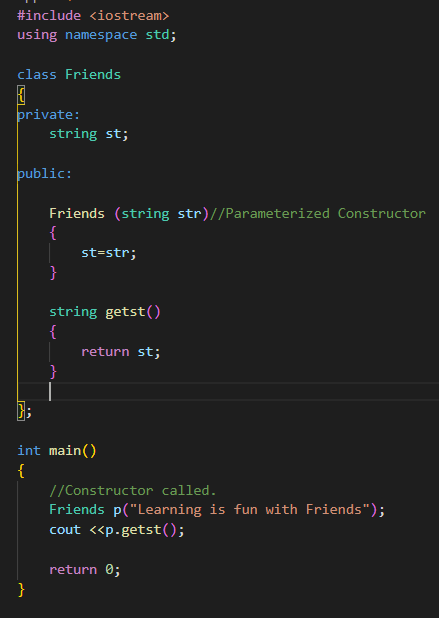


Output:



(b) one argument

Code:

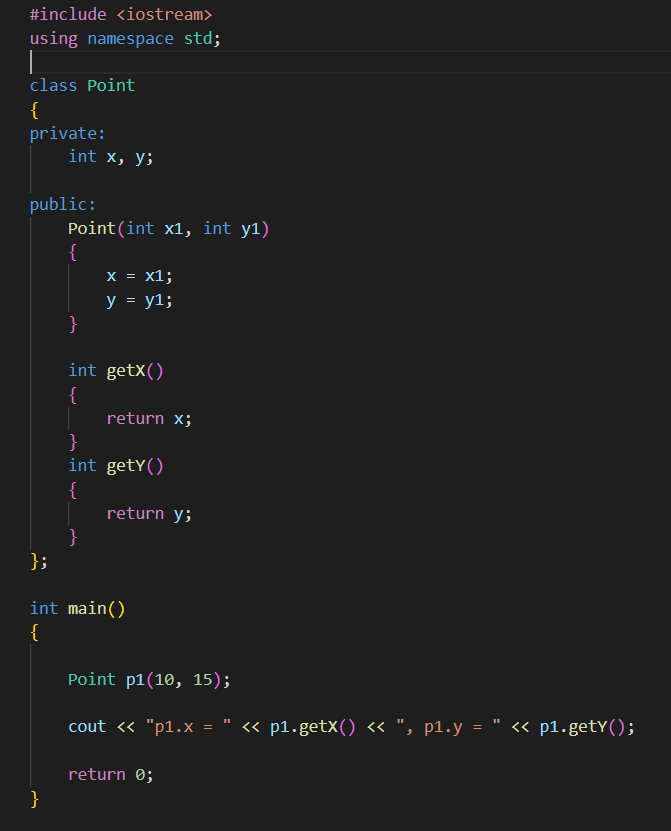


Output:



(2) two arguments

Code:

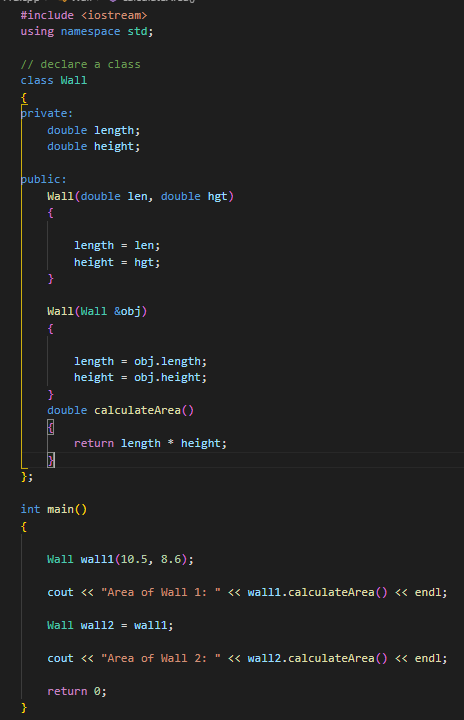


Output:



(3) copy constructor

Code:

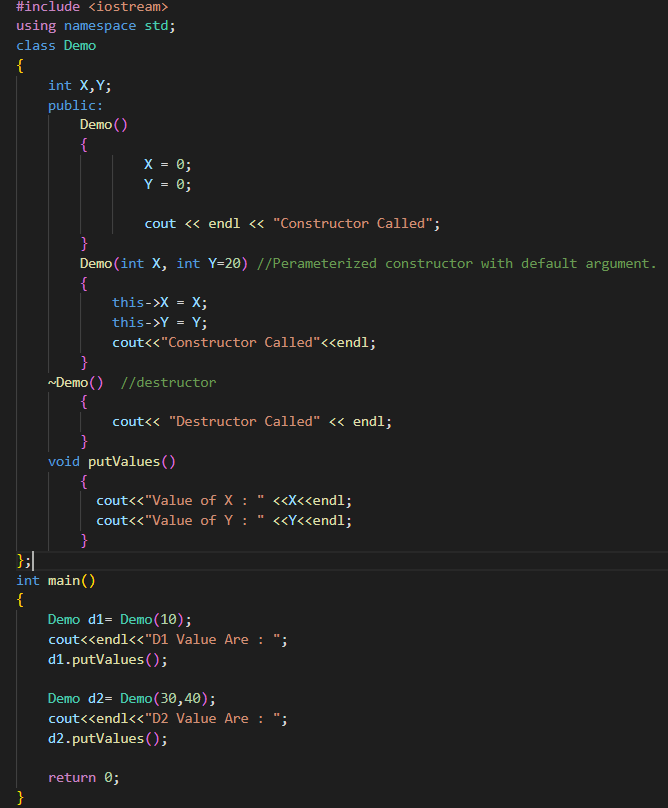


Output:

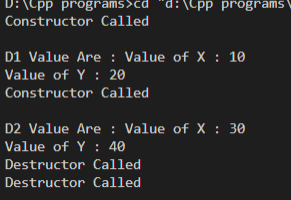


1. Write a program to implement
2. Constructors with default arguments
3. destructor.

Code:

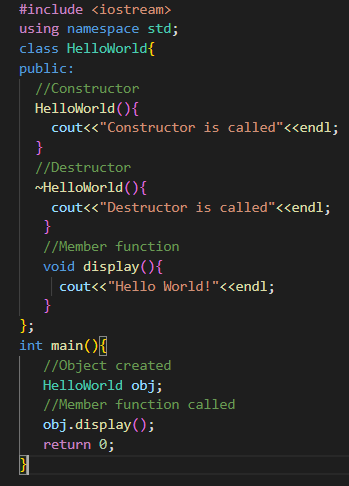


Output:



1. Verify the following about *destructor* by writing the program:
2. Name should begin with tilde sign(~) and must match class name.
3. There cannot be more than one destructor in a class.
4. Destructors do not allow any parameter.
5. They do not have any return type, just like constructors.
6. When you do not specify any destructor in a class, compiler generates a default destructor and inserts it into your code.

Code:



Output:

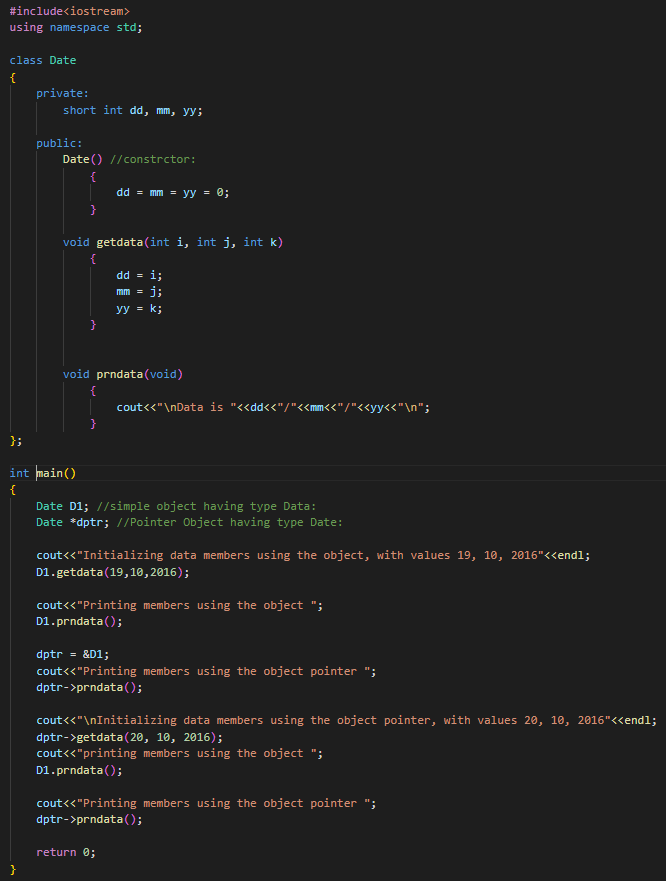


1. Write a program to implement

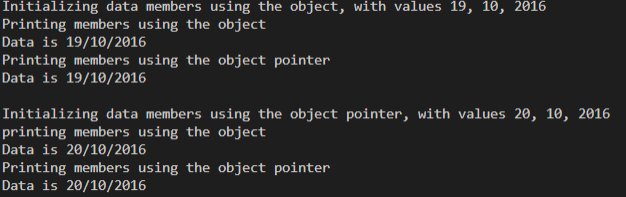
(a) pointer to an objects

(b) this pointer. Practice both ‘.’ (dot operator) and ‘->’ (arrow operator).

Code:



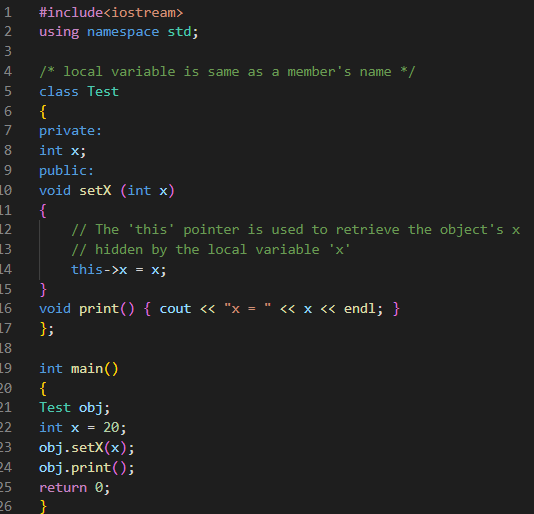
Output:



b)

Code:

Using dot operator:

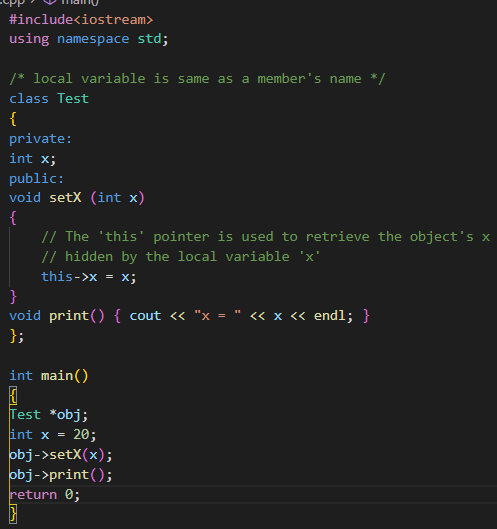


Output:



Code:

Using arrow operator:

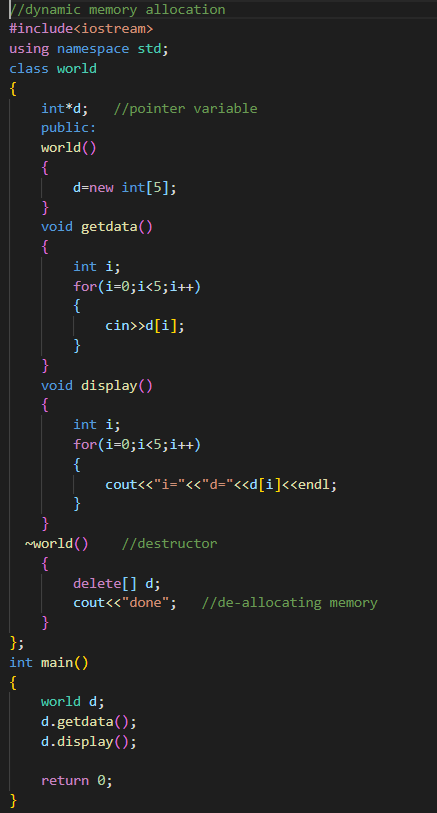


Output:

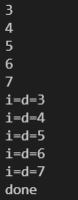


1. How would you implement dynamic memory allocation? Use *new* and *delete* keywords.

Code:

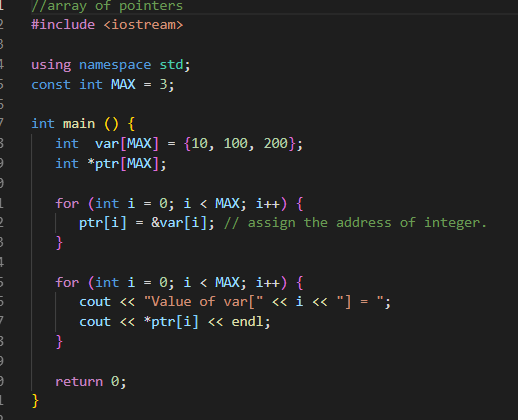


Output:

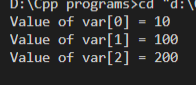


1. Implement array of pointers to the objects in C++.

Code:

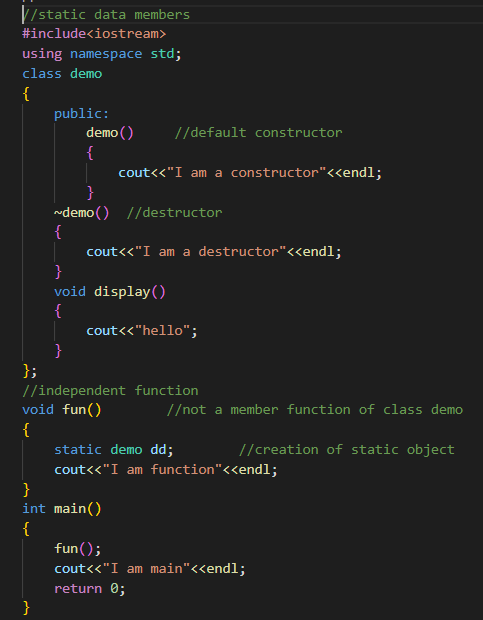


Output:

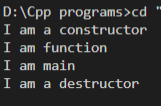


1. Write a program to implement Static objects.

Code:



Output:

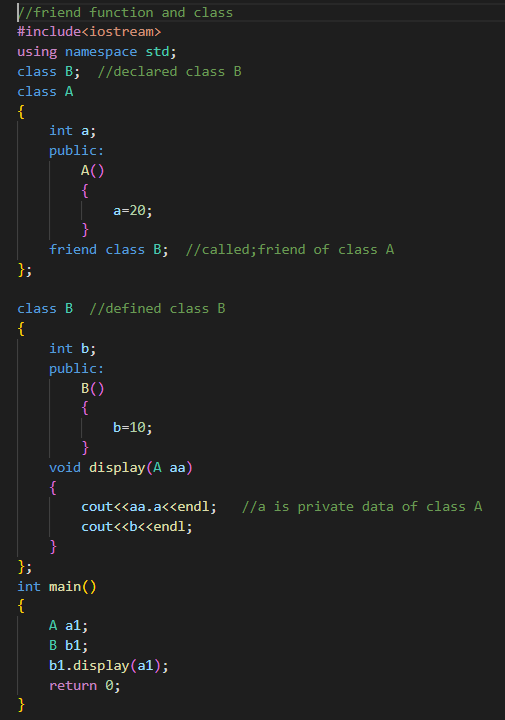


1. Understand the meaning of *friend* keyword and implement

(a) Friend function

(b) Friend class.

Code:



Output:

