1.Constructor

#include<iostream>

using namespace std**;**

class data

**{**

private**:** **int** x**,**y**;**

public**:**

data**()*//Contructor have same name as class***

**{**

x**=**10**;**y**=**20**;**

cout**<<**"Hello i am constructor and i intialize value of x and y as "**<<**x**<<**" and "**<<**y**;**

**}**

**};**

**int** main**()**

**{**

data d**;*//Constructor will be automticaally called***

return 0;

}

2.Constructor

#include<iostream>

using namespace std**;**

class bca

**{**

public**:**

bca**()*//Contructor have same name as class***

**{**

cout**<<**"We are happy to be in bca"**;**

**}**

**};**

**int** main**()**

**{**

bca d**;*//Constructor will be automticaally called***

**return** 0**;**

**}**

3.Constructor

#include<iostream>

using namespace std**;**

class emp\_info**{**

private**:**

**int** salary**,**age**;**

public**:**

emp\_info**(){**

salary**=**70000**;**

age**=**20**;**

**}**

**void** printdata**(){**

cout**<<**"\nEnployee info:\n"**;**

cout**<<**"Employee salary: "**<<**salary**<<**endl**;**

cout**<<**"Employee age: "**<<**age**<<**endl**;**

**}**

**};**

**int** main**()**

**{**

emp\_info i**;**

i**.**printdata**();**

**return** 0**;**

**}**

4.Constructor **(**initializing value**)**

#include<iostream>

using namespace std**;**

class a**{**

**int** num**;**

public**:**

a**();**

show**();**

**};**

a**::**a**()**

**{**

cout**<<**"Enter no: "**;**

cin**>>**num**;**

**}**

**void** a**::**show**()**

**{**

cout"Y0u entered: "**<<**num**;**

**}**

**int** main**()**

**{**

a obj**;**

obj**.**a**();**

obj**.**show**();**

**}**

5.Constructor **(**multiply**)**

#include<iostream>

using namespace std**;**

class multiply

**{**

**int** i**,**x**;**

public**:**

multiply**();**

**void** print\_table**();**

**};**

multiply**::**multiply**()**

**{**

cout **<<** "Enter any number for printing table = "**;**

cin **>>** x**;**

**}**

**void** multiply**::**print\_table**()**

**{**

**for(**i**=**1**;**i**<**11**;**i**++)**

**{**

cout**<<**i**<<**"\*"**<<**x**<<**"="**<<**i**\***x**<<**endl**;**

**}**

**}**

**int** main**()**

**{**

multiply m**;**

m**.**print\_table**();**

**return** 0**;**

**}**

6.Constructor **(**multiply **int** **float)**

#include<iostream>

using namespace std**;**

***// Class section***

class intmult

**{**

private**:** **int** num**;**

public**:**

intmult**();**

printtable1**();**

**};**

***// i/p section member constructor***

intmult**::**intmult**()**

**{**

cout**<<**"\nEnter a integer number:"**;**

cin**>>**num**;**

**}**

***//Member function***

intmult**::**printtable1**()**

**{**

**for(int** i**=**1**;**i**<**11**;**i**++)**

**{**

cout**<<**i**<<**"\*"**<<**num**<<**"="**<<**i**\***num**<<**endl**;**

**}**

**}**

class floatmult

**{**

private**:** **float** num**;**

public**:**

floatmult**();**

printtable2**();**

**};**

floatmult**::**floatmult**()**

**{**

cout**<<**"\nEnter a float number: "**;**

cin**>>**num**;**

**}**

floatmult**::**printtable2**()**

**{**

**for(int** i**=**1**;**i**<**11**;**i**++)**

**{**

cout**<<**i**<<**" \* "**<<**num**<<**" = "**<<**i**\***num**<<**endl**;**

**}**

**}**

**int** main**()**

**{**

intmult obj**;**

obj**.**printtable1**();**

floatmult o**;**

o**.**printtable2**();**

**return** 0**;**

**}**

7.Constructor **(**multiply outside boundaries**)**

#include<iostream>

using namespace std**;**

class multiply

**{**

private**:**

**int** i**,**x**;**

public**:**

multiply**();**

**void** print\_table**();**

**};**

multiply**::**multiply**()**

**{**

cout **<<** "Enter any number for printing table = "**;**

cin **>>** x**;**

**}**

**void** multiply**::** print\_table**()**

**{**

**for(**i**=**1**;**i**<**11**;**i**++)**

**{**

cout**<<**i**<<**"\*"**<<**x**<<**"="**<<**i**\***x**<<**endl**;**

**}**

**}**

**int** main**()**

**{**

multiply m**;**

m**.**print\_table**();**

**return** 0**;**

**}**

8.Constructor **(**multiply within boundaries**)**

#include<iostream>

using namespace std**;**

class multiply

**{**

**int** i**,**x**;**

public**:**

multiply**()**

**{**

cout **<<** "Enter any number for printing table = "**;**

cin **>>** x**;**

**}**

**void** print\_table**()**

**{**

**for(**i**=**1**;**i**<**11**;**i**++)**

**{**

cout**<<**i**<<**"\*"**<<**x**<<**"="**<<**i**\***x**<<**endl**;**

**}**

**}**

**};**

**int** main**()**

**{**

multiply m**;**

m**.**print\_table**();**

**return** 0**;**

**}**

9.Constructor passing arguments

And destructor

#include<iostream>

using namespace std**;**

class add

**{**

public**:**

add**(int** x**,int** y**)**

**{**

cout**<<**"Addition of "**<<**x**<<**" and "**<<**y**<<**" = "**<<**x**+**y**;**

**}**

**void** add\_float**(float** x**,float** y**,float** z**)**

**{**

cout**<<**"\nAddition of "**<<**x**<<**" + "**<<**y**<<**" + "**<<**z**<<**" = "**<<**x**+**y**+**z**;**

**}**

**~**add**()**

**{**

cout**<<**"\nBye bye i am destructor "**;**

**}**

**};**

**int** main**()**

**{**

add obj**(**10**,**20**);**

obj**.**add\_float**(**10.5**,**11.5**,**12.5**);**

**return** 0**;**

**}**

10.Constructor overloading cpp

// C++ program to illustrate

// Constructor overloading

#include <iostream>

using namespace std**;**

class construct

**{**

public**:**float area**;**

// Constructor with no parameters

construct**(**int a**)**

**{**

area **=** a**;**

**}**

// Constructor with two parameters

construct**(**int a**,** int b**)**

**{**

area **=** a **\*** b**;**

**}**

void disp**()**

**{**

cout**<<** area**<<** endl**;**

**}**

**};**

int main**()**

**{**

// Constructor Overloading

// with two different constructors

// of class name

construct o**(**10**);**

construct o2**(** 10**,** 20**);**

o**.**disp**();**

o2**.**disp**();**

**return** 1**;**

**}**

11.Constructor overloading cpp

//Program 2.

#include <iostream>

// Constructor Overloading

using namespace std**;**

class myclass

**{**

int a**,** b**,** c**;**

public**:**

myclass**()**

**{**cout**<<**"Good Afternoon BCA 3 B\n"**;}**

myclass**(**int x**)**

**{**

c**=**x**;**

cout**<<**c**;**

**}**

myclass**(**int i**,** int j**)** **{**a**=**i**;** b**=**j**;}**

void show**()**

**{**

cout **<<** "a="**<<**a **<<** " " **<<**"b="**<<** b**;**

**}**

**};**

int main**()**

**{**

myclass ob**;**

myclass ob1**(**10**);**

myclass ob2**(**3**,**5**);**

ob2**.**show**();**

**return** 0**;**

**}**

12. Parameterized constructor in cpp

#include<iostream>

using namespace std**;**

class add

**{**

public **:** int x**,**y**;**

float a**,**b**;**

add**(**int x**,**int y**)**

**{**

cout**<<**" Addition of "**<<**x**<<**" and " **<<** y **<<**" is = "**<<**x**+**y**<<**endl**;**

**}**

void addfloat**(**float a**,**float b**)**

**{**

cout**<<**" Addition of "**<<**a**<<**" and " **<<**b**<<**" is = "**<<**a**+**b**<<**endl**;**

**}**

**};**

int main**()**

**{**

add obj**(**1**,**2**);**

obj**.**addfloat**(**1.5**,**1.5**);**

**return** 0**;**

**}**

13. Parameterized constructor in cpp

#include<iostream>

using namespace std**;**

class add

**{**

public **:** int x**,**y**;**

float a**,**b**;**

add**(**int x**,**int y**);**

void addfloat**(**float a**,**float b**);**

**};**

add**::**add**(**int x**,**int y**)**

**{**

cout**<<**" Addition of "**<<**x**<<**" and " **<<** y **<<**" is = "**<<**x**+**y**<<**endl**;**

**}**

void add**::**addfloat**(**float a**,**float b**)**

**{**

cout**<<**" Addition of "**<<**a**<<**" and " **<<**b**<<**" is = "**<<**a**+**b**<<**endl**;**

**}**

int main**()**

**{**

add obj**(**1**,**2**);**

obj**.**addfloat**(**1.5**,**1.5**);**

**return** 0**;**

**}**

14.Constructor overloading and copying constructor

#include<iostream>

using namespace std**;**

class test

**{**

**int** a**,**b**;**

public**:**

test**()**

**{**

a**=**10**;**

b**=**20**;**

**}**

test**(int** x**)**

**{**

a**=**x**;**

b**=**40**;**

**}**

test**(int** x**,int** y**)**

**{**

a**=**x**;**

b**=**y**;**

**}**

test**(**test **&**t**)**

**{**

a**=**t**.**a**;**

b**=**t**.**b**;**

**}**

**void** show**()**

**{**

cout**<<**a**<<**b**;**

**}**

**};**

class br

**{**

public**:**

**void** linebreak**()**

**{**

cout**<<**endl**;**

**}**

**};**

**int** main**()**

**{**

test a**;**

a**.**show**();*// 10 20***

br s**;**

s**.**linebreak**();**

test b**(**30**);** ***//30 40***

b**.**show**();**

s**.**linebreak**();**

test c**(**50**,**60**);** ***// 50 60***

c**.**show**();**

s**.**linebreak**();**

test d**(**b**);** ***//30 40***

d**.**show**();**

s**.**linebreak**();**

test e**(**a**);** ***// 10 20***

e**.**show**();**

s**.**linebreak**();**

test d1**(**e**);** ***// 10 20***

d1**.**show**();**

**}**

15.Destructor

#include<iostream>

using namespace std**;**

**int** a**=**0**;**

class test

**{**

public**:**

test**()**

**{**

a**++;**

cout**<<**a**;**

**}**

**~**test**()**

**{**

a**--;**

cout**<<**a**;**

**}**

**};**

**int** main**()**

**{**

test a**,**b**,**c**;**

**{**

test d**;**

**}**

test e**;**

**}**