**EVEREST ENGINEERING COLLEGE**

  **SANEPA, LALITPUR**

(AFFILIATED TO POKHARA UNIVERSITY)

AN

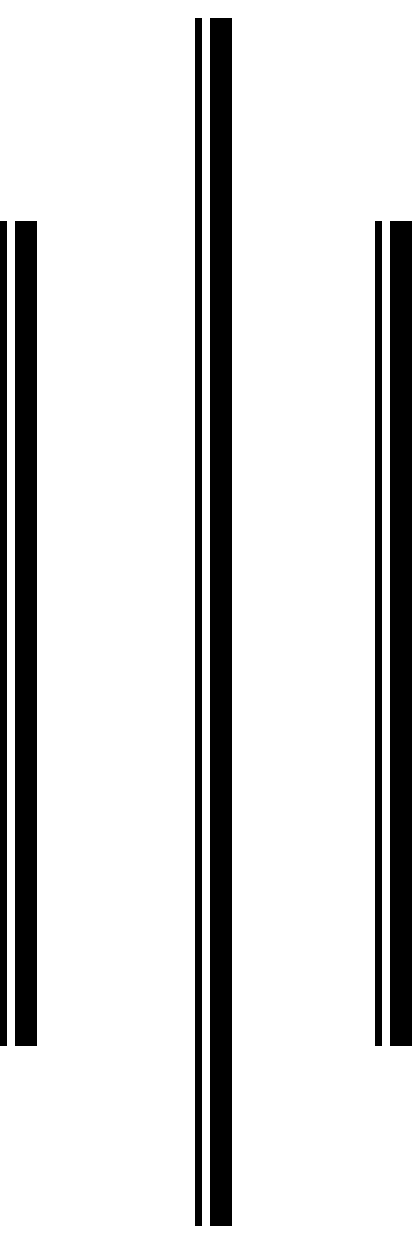
LAB REPORT

ON

**Object Oriented Programming In C++**

***[Constructor & Destructor]***

**Lab Sheet: 5**



**SUBMITTED BY SUBMITTED TO**

Name:-**SAKCHYAM ACHARYA** Er.Pradip Paudel

Roll No:-40 Everest Engineering College

Department of Computer Engineering

Faculty:-BE CMP

Year/Semester: - 2nd SEM

Batch: 2021

Verified By:-

Student's Signature

**Index Page:-**

Topics Page No

* Title, Objective....................................................................................3
* Theory……………………………………………………………………………….3
* Constructor & its types
* Constructor Overloading
* Destructor
* Questions with Source Code & Output…………………………………………...4-14
* Discussion & Conclusion ...................................................................................14

**Title:-Constructor & Destructor.**

**Objective:-**

* To be familiar with constructor and destructor and how to use them.
* To be familiar with constructor overloading.

**Theory:-**

* **Constructor & its types:-**

A constructor is a special member function which initializes the object of its class. It is called special because its name is same as that of the class name.

**Types of Constructor:-**

1. **Default Constructor:-**A constructor that accepts no arguments is called default constructor. If class doesn’t include any constructor, the complier creates default constructor.
2. **Parameterized Constructor:**-A constructor that can take arguments are called parameterized constructor. Arguments are passed when the object are created. This can be done in two ways by:-
3. By calling the constructor explicitly.
4. By calling the constructor implicitly.
5. **Copy Constructor:-** A copy constructor is used to declaring & initialize an object with another object of same type.

* **Constructor Overloading:-**

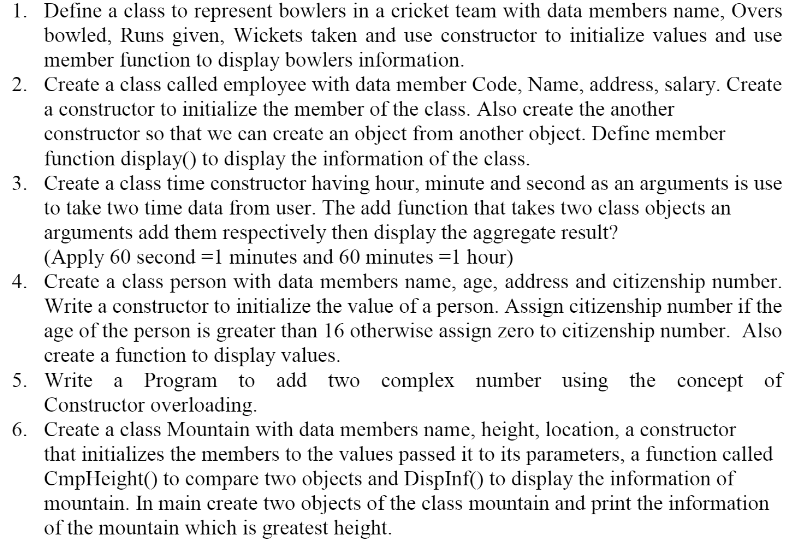
Constructors can be overloaded in a similar way as [function overloading](https://www.programiz.com/cpp-programming/function-overloading).

Overloaded constructors have the same name (name of the class) but the different number of arguments. Depending upon the number and type of arguments passed, the corresponding constructor is called.

* **Destructor:-**

A destructor is a member function that is invoked automatically when the object goes out of scope or is explicitly destroyed by a call to delete.

* **Lab Exercise:-**

****

**///Prob1)**

**//Source Code:**

#include<iostream>

#include<string.h>

using namespace std;

class cricket{

private:

char name[20];

float ov;

int ru,wi;

public:

cricket(char na[],float ove,int run,int wick)

{

strcpy(name,na);

ov=ove;

ru=run;

wi=wick;

}

void displaydata(){

cout<<"Name of player="<<name<<endl;

cout<<"Overs Bowled="<<ov<<endl;

cout<<"Runs="<<ru<<endl;

cout<<"Wicket="<<wi<<endl;

}

};

int main(){

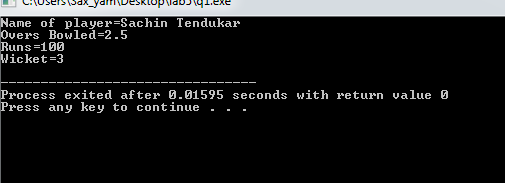
char name[]="Sachin Tendukar";

cricket c1(name,2.5,100,3);

c1.displaydata();

return 0;

}



**///Prob2)**

**//Source Code:**

#include<iostream>

#include<string.h>

using namespace std;

class employee{

private:

int code;

char address[20],name[20];

double salary;

public:

employee(){

}

employee(int co,char add[],char nam[],double sal){

code=co;

strcpy(address,add);

strcpy(name,nam);

salary=sal;

}

employee(employee &x){

code=x.code;

strcpy(name,x.name);

strcpy(address,x.address);

salary=x.salary;

}

void display(){

cout<<"\tDetails of Employee:"<<endl;

cout<<"Code="<<code<<endl;

cout<<"Address="<<address<<endl;

cout<<"Name="<<name<<endl;

cout<<"Salary="<<"Rs."<<salary<<endl;

}

};

int main(){

char name[]="Sujan Lohani(Coder)";

char address[]="Kathmandu";

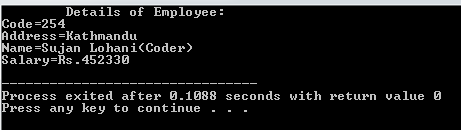
employee e1(254,address,name,452329.89);

employee e2(e1);

e2.display();

return 0;

}



**///Prob3)**

**//Source Code:**

#include<iostream>

using namespace std;

class time{

private:

int hrs,min,sec;

public:

time(){}

time(int h,int m,int s){

hrs=h;

min=m;

sec=s;

}

void addtime(time t1,time t2){

sec=t1.sec + t2.sec;

min=t1.min + t2.min + (sec/60);

hrs=t1.hrs + t2.hrs +(min/60);

min%=60;

sec%=60;

}

void display(){

cout<<hrs<<":"<<min<<":"<<sec<<endl;

}

};

int main(){

int h1,m1,s1,h2,m2,s2;

cout<<"For 1st time"<<endl;

cout<<"Hours1 & Minute1 & Second1:";

cin>>h1>>m1>>s1;

cout<<endl<<"For 2nd time"<<endl;

cout<<"Hours2 & Minute2 & Second2:"<<endl;

cin>>h2>>m2>>s2;

time t1(h1,m1,s1);

time t2(h2,m2,s2);

time t3;

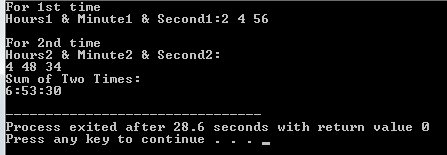
t3.addtime(t1,t2);

cout<<"Sum of Two Times:"<<endl;

t3.display();

return 0;

}



**///Prob4)**

**//Source Code:**

#include<iostream>

#include<string.h>

using namespace std;

class person{

private:

char name[20],address[20];

unsigned int cn;

short int age;

public:

person(){

}

person(char nam[],char addr[],short int ag){

strcpy(address,addr);

strcpy(name,nam);

//cn=crn;

age=ag;

if (age>16){

cn=253637;

}

else{

cn=0;

}

}

void display(){

cout<<endl<<"Name:-"<<name<<endl<<"Address:-"<<address<<endl<<"Age:-"<<age<<endl<<"Citizenship No:-"<<cn<<endl;

//cout<<"\t\tEnter the details:"<<endl;

//cout<<endl<<"Name , Address , CitizenShip No, Age:";

//cin>>name>>address>>cn>>age;

}

};

int main(){

char n[]="Sujit Tamang";

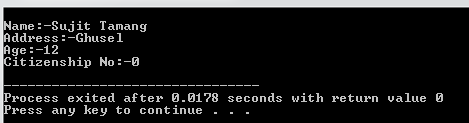
char a[]="Ghusel";

person p1(n,a,12);

p1.display();

return 0;

}



**///Prob5)**

**//Source Code:**

#include<iostream>

using namespace std;

class complex{

private:

int x, y;

public:

complex(int a, int b)

{

x=a;

y=b;

}

complex(){

}

void addcomplex(complex c1,complex c2){

x = c1.x + c2.x;

y = c1.y + c2.y;

}

void display(){

cout<<"Real Part:"<<x<<endl<<"Imaginary Part:"<<y<<"i"<<endl;

cout<<endl<<"Sum of Complex Number="<<x<<"+"<<y<<"i";

}

};

int main(){

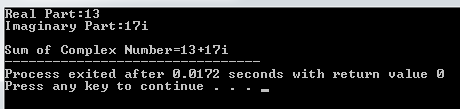
complex c1(7,9),c2(6,8),c3;

c3.addcomplex(c1,c2);

c3.display();

return 0;

}



**///Prob6)**

**//Source Code:**

#include<iostream>

#include<string.h>

using namespace std;

class mountain{

private:

char name[20],loca[20];

double ht;

public:

mountain(char nam[],char locat[],double hei)

{

strcpy(name,nam);

strcpy(loca,locat);

ht=hei;

}

mountain(){};

void cmpheight(mountain m1,mountain m2){

if(m1.ht>m2.ht){

strcpy(name,m1.name);

strcpy(loca,m1.loca);

ht=m1.ht;

}

else{

strcpy(name,m2.name);

strcpy(loca,m2.loca);

ht=m2.ht;

}

}

void display(){

cout<<endl<<"\t\t Mountain Details:"<<endl;

cout<<"Mountain Name:"<<name<<endl<<"Mountain Location:"<<loca<<endl<<"Mountain Height:"<<ht<<endl;

}

};

int main(){

char x[20], y[20];

double hei;

cout<<"Enter Mountain Name:";

cin.getline(x,20);

cout<<"Enter Mountain Location:";

cin.getline(y,20);

//cin.ignore();

cout<<"Enter Mountain Height(in feet):";

cin>>hei;

mountain m1(x,y,hei);

cin.ignore();

cout<<endl<<"Enter Mountain Name:";

cin.get(x,20);

cin.ignore();

cout<<"Enter Mountain Location:";

cin.get(y,20);

cin.ignore();

cout<<"Enter Mountain Height(in feet):";

cin>>hei;

mountain m2(x,y,hei);

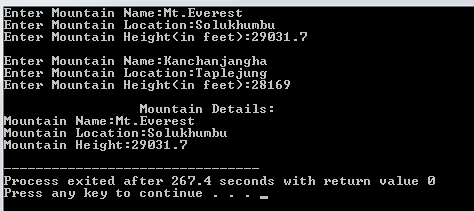
mountain m3;

m3.cmpheight(m1,m2);

m3.display();

return 0;

}



**Discussion & Conclusion:-**

n this lab we have concluded about constructor and destructor , we have concluded how to use

constructor and how to use destructor as we have used in above tasks, with this concept of

construcor and destroctor we can use it with different types of programs .

n this lab we have concluded about constructor and destructor , we have concluded how to use

constructor and how to use destructor as we have used in above tasks, with this concept of

construcor and destroctor we can use it with different types of programs .

In this lab I have concluded about constructor & destructor, I have also learnt constructor overloading & how to use them as we have used in above task.

**Thank You**<SAKWheels>