**Assignment – 26**

**Member function, static, constructor**

1. #include<iostream>

using namespace std;

class complex

{

private:

int a;

int b;

public:

void setData(int real, int img)

{

a=real; b=img;

}

complex add(complex);

void showData();

};

complex complex::add(complex C)

{

complex t;

t.a = a + C.a;

t.b = b + C.b;

return t;

}

void complex::showData()

{

cout<<a<<" + i"<<b<<endl;

}

int main()

{

complex C1,C2,C3;

C1.setData(2.5,7.1);

C2.setData(4.2,5.5);

C3=C1.add(C2);

cout<<"\n complex Number 1 = ";C1.showData();

cout<<"\n complex Number 2 = ";C2.showData();

cout<<"\n complex Number 3 = ";C3.showData();

return 0;

}

2. #include<iostream>

using namespace std;

class Time

{

private:

int h;

int m;

int s;

public:

void setTime(int hour,int min,int sec)

{

h=hour;

m=min;

s=sec;

}

void showTime()

{

cout<<h<<"::"<<m<<"::"<<s<<endl;

}

void normalize()

{

m=m+s/60;

s= s%60;

h= h + m / 60;

m= m%60;

}

};

int main()

{

Time t1;

t1.setTime(5, 125, 130);

t1.showTime();

t1.normalize();

t1.showTime();

return 0;

}

3. #include<iostream>

using namespace std;

class Cube

{

public:

int side;

int volume()

{

return (side\*side\*side);

}

Cube()

{

cout << "\nconstructor is called " << endl;

}

~Cube()

{

cout << "\nDestructing " << side << endl;

}

};

int main()

{

Cube c;

Cube c1;

cout<<"SIDE OF CUBE="<<c1.side<<endl;

cout<<"VOLUME OF FIRST CUBE="<<c1.volume()<<endl;

return 0;

}

4. #include<iostream>

using namespace std;

class counter

{

private:

int count;

public:

counter ()

{

count=0;

}

void inc\_count ()

{

count++;

}

int get\_count ()

{

return count;

}

};

int main ()

{

counter c1;

cout<<"\nBefore calling Counter Function, Count = ";

cout<<c1.get\_count ();

c1.inc\_count ();

cout<<"\n\nAfter calling Counter Function, Count = ";

cout<<c1.get\_count ();

cout<<"\n";

return 0;

}

5. #include<iostream>

using namespace std;

class date

{

private:

int dd, mm, yy;

public:

date()

{

dd=15;

mm=12;

yy=2022;

cout<<"\nDate Object has been created..............\n";

}

void display()

{

cout<<"\nThe Entered Date is :: ";

cout<<dd<<"-"<<mm<<"-"<<yy<<"\n";

}

};

int main ()

{

date date1;

date1.display ();

return 0;

}

6. #include<iostream>

using namespace std;

class student

{

private:

char name[20],add[20];

int roll,zip;

public:

student ();

~student();

void read();

void disp();

};

student :: student()

{

cout<<"\nThis is Student Details constructor called..........."<<endl;

}

void student :: read()

{

cout<<"\nEnter the student Name :: ";

cin>>name;

cout<<"\nEnter the student roll no :: ";

cin>>roll;

cout<<"\nEnter the student address :: ";

cin>>add;

cout<<"\nEnter the Zipcode :: ";

cin>>zip;

}

void student :: disp()

{

cout<<"\nThe Entered Student Details are shown below ::---------- \n";

cout<<"\nStudent Name :: "<<name<<endl;

cout<<"\nRoll no is :: "<<roll<<endl;

cout<<"\nAddress is :: "<<add<<endl;

cout<<"\nZipcode is :: "<<zip;

}

student :: ~student()

{

cout<<"\n\nStudent Detail is Closed.............\n";

}

int main()

{

student s;

s.read ();

s.disp ();

return 0;

}

7. #include<iostream>

using namespace std;

class Box

{

private:

int length;

int bredth;

int height;

int vol()

{

return(length\*bredth\*height);

}

public:

Box();

~Box();

void read();

void disp();

};

Box::Box()

{

cout<<"Constructor is called"<<endl;

}

void Box::read()

{

cout<<"ENTER LENGTH=";

cin>>length;

cout<<"\nENTER BREADTH=";

cin>>bredth;

cout<<"\nENTER HEIGTH=";

cin>>height;

cout<<endl;

}

void Box::disp()

{

cout<<"LENGTH OF BOX="<<length<<endl;

cout<<"BREADTH OF BOX="<<bredth<<endl;

cout<<"HEIGHT OF BOX="<<height<<endl;

cout<<"VOLUME OF BOX="<<vol()<<endl;

}

Box::~Box()

{

cout<<"Destructor is called"<<endl;

}

int main()

{

Box b;

b.read();

b.disp();

return 0;

}

8. #include<iostream>

#include<string.h>

using namespace std;

class Bank

{

private:

float p;

float roi;

int y;

float si()

{

return((p\*roi\*y)/100);

}

public:

Bank();

~Bank();

void read();

void disp();

};

Bank::Bank()

{

cout<<"Constructor is called"<<endl;

}

void Bank::read()

{

cout<<"ENTER PRINCIPLE AMOUNT=";

cin>>p;

cout<<"\nRATE OF INTREST=";

cin>>roi;

cout<<"YEAR=";

cin>>y;

cout<<"\n-------------------------------------------------------------------------------"<<endl;

}

void Bank::disp()

{

cout<<"Simple intrest="<<si()<<endl;

}

Bank::~Bank()

{

cout<<"Destructor is called"<<endl;

}

int main()

{

Bank b;

b.read();

b.disp();

return 0;

}

9. #include<iostream>

using namespace std;

class Bill

{

private:

int unit;

public:

Bill();

~Bill();

void get();

void cal();

void disp();

};

Bill::Bill()

{

cout<<"CONSTRUCTOR called"<<endl;

}

void Bill::get()

{

cout<<"ENTER UNIT=";

cin>>unit;

cout<<endl;

}

void Bill::cal()

{

if(unit<=100)

{

cout<<"BILL GENERATED AMOUNT OF Rs:"<<unit\*1.20<<endl;

}

else if(unit>100 && unit<=200)

{

cout<<"BILL GENERATED AMOUNT OF Rs:"<<unit\*2<<endl;

}

else if(unit>200)

{

cout<<"BILL GENERATED AMOUNT OF Rs:"<<unit\*3<<endl;

}

}

Bill::~Bill()

{

cout<<"Destructor is called"<<endl;

}

int main()

{

Bill b;

b.get();

b.cal();

return 0;

}

10. #include <iostream>

#include <string>

using namespace std;

void demo()

{

static int count = 0;

cout << count << " ";

count++;

}

int main()

{

for (int i=0; i<3; i++)

demo();

return 0;

}