**Assignment - 26 A Job Ready Bootcamp in C++, DSA and IOT MySirG**

**Member function, static, constructor**

1. Define a class Complex to represent a complex number with instance variables a and

b to store real and imaginary parts. Also define following member functions

a. void setData(int,int)

b. void showData()

c. Complex add(Complex)

Sol – 1.

#include<iostream>

using namespace std;

class Complex

{

private :

int a,b;

public :

void setval(int real, int img)

{

a=real;

b=img;

}

void print()

{

cout<<"Complex no. : Real "<<a<<" and img "<<b;

}

Complex add(Complex C)

{

Complex temp;

temp.a=a+C.a;

temp.b=b+C.b;

return temp;

}

};

int main()

{

int r,i;

Complex n1,n2,n3;

cout<<"Enter real and imag part : ";

cin>>r>>i;

n1.setval(r,i);

cout<<"Enter real and imag part : ";

cin>>r>>i;

n2.setval(r,i);

n3=n1.add(n2);

n3.print();

return 0;

}

2. Define a class Time to represent a time with instance variables h,m and s to store

hour, minute and second. Also define following member functions

a. void setTime(int,int,int)

b. void showTime()

c. void normalize()

d. Time add(Time)

Sol – 2.

#include<iostream>

using namespace std;

class Time

{

private :

int a,b,c;

public :

void setval(int hr, int min,int sec)

{

a=hr;

b=min;

c=sec;

}

void showTime()

{

cout<<"Time : "<<a<<" hr "<<b<<" min "<<c<<" sec";

}

void normalize()

{

/\*while(c>59)

{

c=c-60;

b++;

}

while(b>59)

{

b=b-60;

a++;

}

OR\*/

b=b+c/60;

c=c%60;

a=a+b/60;

b=b%60;

}

Time add(Time T)

{

Time temp;

temp.a=a+T.a;

temp.b=b+T.b;

temp.c=c+T.c;

return temp;

}

};

int main()

{

int h,m,s;

Time n1,n2,n3;

cout<<"Enter hours minutes and seconds : ";

cin>>h>>m>>s;

n1.setval(h,m,s);

cout<<"Enter hours minutes and seconds : ";

cin>>h>>m>>s;

n2.setval(h,m,s);

n3=n2.add(n1);

n3.normalize();

n3.showTime();

return 0;

}

3. Define a class Cube and calculate Volume of Cube and initialise it using constructor.

Sol – 3.

#include<iostream>

using namespace std;

class Cube

{

private :

int a;

public :

Cube(int x){ a=x; }

int getside()

{

return a;

}

int vol()

{

return (a\*a\*a);

}

};

int main()

{

int a;

cout<<"Enter side of cube : ";

cin>>a;

Cube n1(a);

cout<<"Volume : "<<n1.vol();

return 0;

}

4. Define a class Counter and Write a program to Show Counter using Constructor.

Sol – 4.

#include<iostream>

using namespace std;

class Counter

{

static int count;

public :

Counter()

{

count++;

}

int getcount()

{

return count;

}

};

int Counter::count;

int main()

{

Counter a,b,c;

cout<<a.getcount();

return 0;

}

OR

#include<iostream>

using namespace std;

class Counter

{

int count;

public :

Counter()

{

count=0;

}

void counting()

{

count++;

}

int getcount()

{

return count;

}

};

int main()

{

Counter a,b,c;

a.counting();

a.counting();

cout<<a.getcount();

return 0;

}

5. Define a class Date and write a program to Display Date and initialise date object

using Constructors.

Sol – 5.

#include<iostream>

using namespace std;

class Date

{

private :

int d,m,y;

public :

Date(int date,int month,int year)

{

d=date;

m=month;

y=year;

}

void display()

{

cout<<"Date : "<<d<<"/"<<m<<"/"<<y;

}

};

int main()

{

Date a(06,10,2022);//today's date

a.display();

return 0;

}

6. Define a class student and write a program to enter student details using constructor

and define member function to display all the details.

Sol – 6.

#include<iostream>

#include<string.h>

using namespace std;

class Student

{

int r,m,c;

char n[20];

public :

Student(char\* name,int cass,int marks,int rollno)

{

strcpy(n,name);

c=cass;

r=rollno;

m=marks;

}

void display()

{

cout<<"Name : "<<n<<"\nClass : "<<c<<"\nRoll no : "<<r<<"\nMarks : "<<m;

}

};

int main()

{

Student n1("Student",12,430,2);

n1.display();

return 0;

}

7. Define a class Box and write a program to enter length, breadth and height and

initialise objects using constructor also define member functions to calculate volume

of the box.

Sol – 7.

#include<iostream>

using namespace std;

class Box

{

private :

int l,b,h;

public :

Box(int length,int breadth,int height){ l=length;b=breadth;h=height; }

int vol()

{

return (l\*b\*h);

}

};

int main()

{

int a;

Box n1(5,10,3);

cout<<"Volume : "<<n1.vol();

return 0;

}

8. Define a class Bank and define member functions to read principal , rate of interest

and year. Another member functions to calculate simple interest and display it.

Initialise all details using constructor.

Sol – 8.

#include<iostream>

using namespace std;

class Bank

{

private :

int p,r,t;

float si;

public :

Bank(int principal,int rate,int time)

{

p=principal;

r=rate;

t=time;

}

void simpleint()

{

si=(p\*r\*t)/100.0;

cout<<"Simple Interest is "<<si;

}

};

int main()

{

Bank b1(800,5,1);

b1.simpleint();

return 0;

}

9. Define a class Bill and define its member function get() to take detail of customer ,

calculateBill() function to calculate electricity bill using below tariff :

Upto 100 unit RS. 1.20 per unit

From 100 to 200 unit RS. 2 per unit

Above 200 units RS. 3 per unit.

Sol – 9.

#include<iostream>

#include<string.h>

using namespace std;

class Bill

{

private :

int unit,house;

char nam[20],ct[20];

float bill;

public :

void setval(int houseno,int units,char \*name,char \*city)

{

unit=units;

house=houseno;

strcpy(nam,name);

strcpy(ct,city);

}

void calculate\_bill()

{

if(unit<=100)

{

bill=unit\*1.20;

}

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

{

bill=(100\*1.20)+((unit-100)\*2);

}

else

{

bill=120+200+((unit-200)\*3);

}

}

void showdata()

{

cout<<"Name - "<<nam<<"\nHouse No. "<<house<<"\nCity "<<ct;

}

float getbill()

{

return bill;

}

};

int main()

{

Bill n1;

n1.setval(125,323,"Owner","City");

n1.showdata();

n1.calculate\_bill();

cout<<"\nBill : "<<n1.getbill()<<" INR";

return 0;

}

10. Define a class StaticCount and create a static variable. Increment this variable in a

function and call this 3 times and display the result.

Sol – 10.

#include<iostream>

#include<string.h>

using namespace std;

class StaticCount

{

private :

static int x;

public :

void statinc()

{

x++;

}

int getval()

{

return x;

}

};

int StaticCount::x=0;

int main()

{

StaticCount A;

A.statinc();

A.statinc();

cout<<A.getval();

return 0;

}