CHAUDHARY HAMDAN

1905387
OOP LAB-4
Date : 21-08-2020

1.WAP to find area of a circle,a rectangle and a triangle using concept of function overloading.

```
#include<iostream>
using namespace std;
int main()
{
	int area(int);
	double area(int,double);
	int area(int,int);
	cout<<area(2)<<endl;
	cout<<area(2,3.0)<<endl;
	cout<<area(2,3)<<endl;
	return 0;
}
int area(int p)
{
	return (3.14*p*p);
}
double area(int p,double q)
{
	return (p*q);
}
int area(int p,int q)
{
	return (0.5*p*q);
}
```

OUTPUT:

12

6

3

2.WAP to find volume of a sphere, a cylinder and a cuboid using function overloading.

```
#include<iostream>
using namespace std;
int main()
{
  int vol(int);
  double vol(int ,double);
  int vol(int ,int, double);

  cout<<vol(3)<<endl;
  cout<<vol(3,6.0)<<endl;
  cout<<vol(3,6,9.0)<<endl;

return 0;
}

int vol(int a)
{
  int s=4*22*a*a*a;
  return (s/21);
}

double vol(int p,double q)
{
  int r=22*p*p*q;
  return (r/7);
}

int vol(int c,int d,double e)
{
  return (c*d*e);
}
```

OUTPUT:

113
169
162

3.WAP which displays a given character,n no of times using a function.When the n value is not provided, it should print the given character 80 times.When both the character and n value is not provided, it should print'*' character 80 times.

```
#include<iostream>
using namespace std;
void num(char c='*',int n=80)
{
        while (n>0)
        {
        cout<<c;
        n--;
        }
        cout<<endl;
}
int main()
{
        num('A',6);
        num('A');
        num();
return 0;
}
```

OUTPUT:

AAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

AAAAAA

4.WAP to find square and cube of a number using inline functions.

```
#include<iostream>
using namespace std;
inline int square(int n)
{
	return (n*n);
}
inline int cube(int n)
{
	return (n*n*n);
}
int main()
{
	cout<<square(3)<<endl;
	cout<<cube(3)<<endl;
return 0;
}
```

OUTPUT:

```
9
27
```

5.WAP to swap two integers using pass by reference.

```
#include<iostream>
using namespace std;
int main()
{
	void swap(int &,int &);
	int a,b;
	cout<<"Enter the numbers:"<<endl;
	cin>>a>>b;
	swap(a,b);
return 0;
}
void swap(int &p,int &q)
{
	int s;
	s=p;
	p=q;
	q=s;
	cout<<"a ="<<p<<endl<<"b ="<<q<<endl;
}
```

OUTPUT:

```
Enter the numbers:
12
3
a =3
b =12
```

6.WAP to increment the value of an argument given to function.

```
#include<iostream>
using namespace std;
int main()
{
void inc(int);
	int n;
	cin>>n;
	inc(n);
return 0;
}
void inc(int a)
{
	a=a+1;
	cout<<a<<endl;
}
```

OUTPUT:

```
9
10
```

7.WAP to display no. From 1 to 10 by using static data member.

```
#include<iostream>
using namespace std;
class number
{
	static int n;
	public:
	void getdata()
	{
	cin>>n;
	}
	void display()
	{
	for (int i=1;i<=n;i++)
	{
	cout<<i<<" ";
	}
	}
};
int number::n;
int main()
{
	number c;
	c.getdata();
	c.display();
return 0;
}
```

OUTPUT:
10

1 2 3 4 5 6 7 8 9 10

8.WAP to find simple interest by using static data member and member function.

```
#include<iostream>
using namespace std;
class simple
{
	static float p;
	static int r;
	static int y;
	public:
	void getdata()
	{
	cin>>p>>r>>y;
	}
	static void calculate()
	{
	float SI;
	SI=p*r*y*0.01;
	cout<<SI;
	}
};
float simple::p;
int simple::r;
int simple::y;
int main()
{
	simple ob;
	ob.getdata();
	simple::calculate();
return 0;
}
```

OUTPUT:
12000
10
3
3600

9.WAP to display to enter student roll,name,mark and display their cgpa by using static member function.

```
#include<iostream>
using namespace std;
class student
{
	static int roll;
	static char name[30];
	static float marks[3];
	static float cgpa;
	static float sum;
	public:
	void getdata()
	{
	cout<<"Enter the name: ";
	cin>>name;
	cout<<"Enter the roll: ";
	cin>>roll;
	cout<<"Enter the marks: ";
	for (int i=0;i<3;i++)
	{
	cin>>marks[i];
	}
	}
	static void display()
	{
	for (int i=0;i<3;i++)
	{
	sum=sum+marks[i];
	}
	cgpa=(sum/3)/9.5;
	cout<<"Name: "<<name<<endl;
	cout<<"Roll: "<<roll<<endl;
```

```
        cout<<"Cgpa: "<<cgpa<<endl;
        }
};
int student::roll;
char student::name[30];
float student::marks[3];
float student::cgpa;
float student::sum;
int main()
{
        student ob;
        ob.getdata();
        student::display();
return 0;
}
```

OUTPUT:

```
Enter the name: KIITian
Enter the roll: 1905000
Enter the marks: 12
13
15
Name: KIITian
Roll: 1905000
Cgpa: 1.40351
```

10.WAP to over load area(circle,rectangle)function by using static member function.

```
#include<iostream>
using namespace std;
class A
{
	static float areac,arear;
	public:
	 static int r,l,b;
	void read()
	{
	cout<<"Enter the radius: \n";
	cin>>r;
	cout<<"Enter the length: \n";
	cin>>l;
	cout<<"Enter the breadth: \n";
	cin>>b;
	}
	static void area(int r)
	{
	   areac=3.14*r*r;
	}
	static void area(int l, int b)
	{
	   arear=l*b;
	}

	void display()
	{
	cout<<"Area of circle: "<<areac<<endl;
	cout<<"Area of rectangle: "<<arear<<endl;
	}
};
```

```
int A::r;
int A::l;
int A::b;
float A::areac;
float A::arear;
int main()
{
        A ob;
        ob.read();
        A::area(A::r);
        A::area(A::l,A::b);
        ob.display();
    return 0;
}
```

OUTPUT:

```
Enter the radius: 10
Enter the length: 20
Enter the breadth: 30
Area of circle: 314
Area of rectangle: 600
```

11.WAP in c++ to design a class Employee having data member empid,ename,basic,TA,DA and Gross.Create a static data member bonus shared by all employees.
Use read() member function to read data.
Use show() member function to display data.
Use calc() member function to calculate TA=30%of basic and DA=0%of basic and Gross salary.Read data for n employees.

```
#include<iostream>
using namespace std;
class Employee
{
	int empid;
	char ename[30];
	float basic;
	float TA;
	float DA;
	float gross;
	static float bonus;
	public:
	void read()
	{
	cout<<"Enter id: ";
	cin>>empid;
	cout<<"Enter name: ";
	cin>>ename;
	cout<<"Enter basic: ";
	cin>>basic;
	cout<<"Enter bonus: ";
	cin>>bonus;
	}
	void show()
	{
	cout<<"Name :"<<ename<<endl;
	cout<<"Id :"<<empid<<endl;
```

```
        cout<<"Basic :"<<basic<<endl;
        cout<<"TA :"<<TA<<endl;
        cout<<"DA :"<<DA<<endl;
        cout<<"Bonus :"<<bonus<<endl;
        cout<<"Gross :"<<gross<<endl;
        }
        void calc()
        {
        TA=30*basic*0.01;
        DA=70*basic*0.01;
        gross=TA+DA+basic+bonus;
        }
};
float Employee::bonus;
int main()
{
        int n;
        cin>>n;
        Employee e[n];
        for (int i=0;i<n;i++)
        {
        e[i].read();
        }
        for (int i=0;i<n;i++)
        {
        e[i].calc();
        e[i].show();
        }
return 0;
}
```

OUTPUT:
3
Enter id: 1905001
Enter name: man1
Enter basic: 12000
Enter bonus: 3000
Enter id: 1905002
Enter name: man2
Enter basic: 30000
Enter bonus: 15000
Enter id: 1905003
Enter name: man3
Enter basic: 17000
Enter bonus: 3000

Name :man1
Id :1905000
Basic :12000
TA :3600
DA :8400
Bonus :3000
Gross :27000

Name :man2
Id :1905002
Basic :30000
TA :9000
DA :21000
Bonus :3000
Gross :63000

Name :man3
Id :1905003
Basic :17000
TA :5100
DA :11900
Bonus :3000
Gross :37000

12.WAP in c++ to design a class complex having data member real and imaginary.Use apppropriate member function to perform i/o operations.
Create two objects and add them.

```
#include<iostream>
using namespace std;
class complex
{
	int real;
	int imag;
	public:
	void input()
	{
	cout<<"Enter real part: ";
	cin>>real;
	cout<<"Enter imaginary part: ";
	cin>>imag;
	}
	void output(complex S,complex A)
	{
	real=S.real+A.real;
	imag=S.imaginary+A.imag;
	cout<<"Sum is: "<<sumreal<<"+i"<<sumimag<<endl;
	}
};
int main()
{
	complex ob1,ob2;
	ob1.input();
	ob2.input();
	ob1.output(ob1,ob2);
return 0;
}
```

OUTPUT:
Enter real part: 3
Enter imaginary part: 6
Enter real part: 5
Enter imaginary part: 6
Sum is: 8+i12

13.WAP in c++ to design a class complex having data member real and imaginary.Use appropriate member function to perform i/o operations. Create two objects and add them and store the result in the third object.

```
#include<iostream>
using namespace std;
class complex
{
	int real;
	int imag;
	public:
	void input()
	{
	cout<<"Enter real part: ";
	cin>>real;
	cout<<"Enter imaginary part: ";
	cin>>imag;
	}
	void sum(complex S,complex A)
	{
	real=S.real+A.real;
	imag=S.imag+A.imag;
	}
	void output()
	{
	cout<<real<<"+ i"<<imag<<endl;
	}
};
int main()
{
	complex ob1,ob2,ob3;
	ob1.input();
	ob2.input();
	ob3.sum(ob1,ob2);
	ob3.output();
return 0;
}
```

OUTPUT:

```
Enter real part: 3
Enter imaginary part: 6
Enter real part: 1
Enter imaginary part: 5
4+ i11
```