Date : / /
Page No. : 8

Practical Set - 2

* Write a program to check whether given number is the prime number or not. using a member function.

```
#include<iostream>
using namespace std;
int main()
{
 int i, n, count=0;

 cout<<"Enter the number"<<endl;
 cin>>n;

 for(i=1; i<=n; i++)
 {
  if(n%i==0)
  {
   count++;
  }
 }

 if(count==2)
 {
  cout<<"Number is prime"<<endl;
 }
 else
 {
  cout<<"Number is not prime"<<endl;
 }
 return 0;
}
```

=> Output:

-> Enter the number
8
Number is not prime.

-> Enter the number
2
Number is prime.

Date : / /
Page No. : 9

7 Write a program to find out the maximum number in a given array.

```
#include <iostream>
using namespace std;
int main ()
{ int a[5], i, max, min;
  for (i=0; i<5; i++)
  {
   cout << "Enter the number" << endl;
   cin >> a[i];
  }
  min = a[0];
  max = a[0];

  for (i=0; i<5; i++)
  {
    if (a[i] < min)
      min = a[i];
    if (a[i] > max)
      max = a[i]
  }
  cout << "min number" << min << endl;
  cout << "max number" << max << endl;
  return 0;
}
```

=> Output:

-> Enter the number
21
Enter the number
22
Enter the number
23
Enter the number
24
Enter the number
25

min number 21
max number 25

Practical Set - 3

Page No.: 19 GOPI
Date: / /

8. Write a program to demostrate the use of inline function by creating 2 inline function. one inline for multiplication operation and other inline for division operation.

```
#include <iostream>
using namespace std;

inline void multiplication (int a, int b)
{
 cout << "The multiplication of no: is " << a * b << endl;
}

inline void division (float a, float b)
{
 cout << " The division of no: is" << a/b << endl;
}

int main ()
{
 int c, d;
 cout << "Enter the value of " << endl;
 cin >> c;
 cout << "Enter the value of " << endl;
 cin >> d;

 multiplication (c, d);
 division (c, d);

 return 0;
}
```

=> Output:

```
Enter the value of
4
Enter the value of
3

The multiplication of no: 12
The division of no: 1.3333
```

4 Write a function power to raise a number m to power n. The function takes a double value for m and int value for n. Use default value for n to make the function to calculate squares when this argument is omitted.

```
#include <iostream>
using namespace std;
double power (double m, int n=2)
{
  int i, multiplication;
  for (i=0; i<n; i++)
  {
   multiplication = multiplication * m;
  }
   return multiplication;
}
  int main()
{ int m, n;
 {
  cout << "Enter the value" <<endl;
  cin >> m;
  cout << "Enter the value" <<endl;
  cin >> n;

  cout << "without default value" <<power(m, n)
                                  <<endl;

  cout << "with default value" <<power(m)
                               <<endl;
 }
 return 0;
}
```

=> Output:

```
Enter the value
3
Enter the value
4
without default value 81
with default value 9
```

10. Write a program that overloads absolute 3 functions namely absolute that return absolute values of int, floating point and double number.

```
#include <iostream>
using namespace std;

void absoluate (int i)
{
 cout << i <<endl;
}

void absoluate (double d)
{
 cout << d << endl;
}

void absoluate (float f)
{
 cout << f << endl;
}

int main()
{
 absoluate (10);
 absoluate (10.10);
 absoluate ('ab');
 absoluate (23.006);

 return 0;
}
```

=> Output:

```
10
10.1
24930
23.006
```

Write a program that overloads volume function that return volume of cube, cuboids and cylinder

```
#include <iostream>
using namespace std;
int l, b, h, r;

inline void volume(int l)
{
cout<<"Volume of cube"<<l*l*l<<endl;
}
inline void volume(double l, double b, double h)
{
cout<<"Volume of cuboids"<<l*b*h<<endl;
}
inline void volume(float r, float h)
{
cout<<"Volume of cylinder"<<3.14*r*r*h
<<endl;
}

int main()

{
int l, b, h, r;
cout<<"Enter the value of l"<<endl;
cin>>l;
cout<<"Enter the value of b"<<endl;
cin>>b;
cout<<"Enter the value of h"<<endl;
cin>>h;
```

```
cout<<"Enter the value of r"<<endl;
cin>>r;

volume();
volume(l, b, h);
volume(r, h);

return 0;

}
```

Practical
27/04/22

=> Output:

```
Enter the value of l
2
Enter the value of b
3
Enter the value of h
4
Enter the value of r
5

volume of cube 8
volume of cuboid 24
volume of cylinder 314
```

Date : / /
Page No. :

Practical Set-4

12 Write a program to create a class for book having title, price and publisher having 2 member function getdetail() and setdetail(). Also create the object of 2 different books.

```
#include <iostream>
 using namespace std;
 class book
 {
  public:
      string title;
      string publisher;
      string price;

   int setdata()
   {
    cout<<"Enter the book name"<<endl;
    cin >> title;
    cout <<" Price of book"<<endl;
    cin >> price;
   }

   int getdata()
   {
    cout <<" The book name :"<<title<<endl;
    cout<<"Book price"<<price<<endl;
   }
 };
```

Date : / /
Page No. :

```
int main()
{
  book ob1, ob2;

  ob1.setdata();
  ob2.setdata();

  ob1.getdata();
  ob2.getdata();

  return 0;

}
```

Enter the book name
Maths
Price of book
200
Enter the book name
Maths 2
Price of book
300
The book name : Maths
Book price : 200
The book name : Maths 2
Book price : 300

Date :
Page No. :

13 Write a program to create an array of 5 objects for previous program.

```
#include <iostream>
using namespace std;
class book
{
 public:
    string name;
    string p;
 int setdata()
 {
  cout<<"Enter the name"<<endl;
  cin>>name;
  cout<<"Price of book"<<endl;
  cin>>p;
 }
  int getdata()
 {
   cout<<"Book name:"<<name<<endl;
   cout<<"Price of book:"<<p<<endl;
 }

};

const int data=5;
int main()
```

Date :
Page No. :

```
{
  int i;
  book obj[data];

  For(i=0; i<data; i++)
  {
    cout<<"Data"<<endl;
    obj[i].setdata();
    obj[i].getdata();
  }

  return 0;

}
```

Page No. ____
__/__/__

data
=> Enter the name
Maths
Price of book
100
Enter the name
Maths 1
Price of book
200

data

Enter the name
Maths 2
Price of book
300

data

Enter the name
Maths 3
Price of book
400

data
Enter the name
Maths 4
Price of book
500

-> Book name
Maths
Price of book
100

Book name
Maths 1
Price of book
200

Book name
Maths 2

Price of book
300

Book name
Maths 3

Price of book
400

Book name
Maths 4
Price of book
500

Date : / /
Page No. :

14 Write a ~~pb~~ program to find the maximum price of 2 books and return that book as object. Use object as function argument.

```
#include <iostream>
 using namespace std;
 class book
 {
   public:
       string name;
       string p;

    int setdata()
    {
     cout<<"Enter the name"<<endl;
     cin>>name;
     cout<<" Price of book"<<endl;
     cin>>p;
    }
    int maximumprice(book ob1, book ob2)
    {
     if(ob1.p > ob2.p)
     {
      cout<<" Book 1 price is maximum"<<endl;
     }
     else
     {
      cout<<"Book 2 price is maximum"<<endl;
};};
```

Date : / /
Page No. :

```
int main()
{
 book ob1, ob2, ob3;

 ob1.setdata();
 ob2.setdata();
 ob3.maximumprice(ob1, ob2);

 return 0;
}
```

Date:

Enter the name
Maths
Price of book
200

Enter the name
Maths 1
Price of book
300

Book 2 price is maximum.

Date : / /
Page No. :

15 With reference to previous program, add 2 number variable serial no and no of book in class named books. Create a friend function getcount() to demostrate the concept of Friend function.

```
#include <iostream>
using namespace std;
class book

{
  public:
        string name;
        static int serial_no;
        static int no_of_book;

   void setdata()
   {
     no_of_book++;
     cout<<"Enter the serial no"<<endl;
     cin>>serial_no;
     cout<<"Enter the book name"<<endl;
     cin>>name;
   }

    Void getdata()
  {
```

```
    cout<<"Serial no -->"<<serial_no<<endl;
    cout<<"Book name -->"<<name<<endl;
  }

  friend void getcount();

  void getcount()
  {
    cout<<"Number of book-->"<<book::no_of_book
                              <<endl;

  }
};

int book::serial_no;
int book::no_of_book;

int main()
{
  book b1, b2;
  b1.setdata();
  b1.getdata();
  b1.getcount();
  b2.setdata();
  b2.getdata();
  b2.getcount();

  return 0;
}
```

=> Enter the serial no

23

Enter the book name

Maths

Enter the serial no

24

Enter the book name

Maths 1

Serial no --> 23
Book name --> Maths

Serial no --> 24
Book name --> Maths 1

Number of book --> 2

Date : / /
Page No. :

16 Write a program to demonstrate the use of scope resolution.

```
#include<iostream>
using namespace std;
class student
{
 public:
    int n;
    string name;
    string branch;
  void setdata();
  void getdata();
};

void student::setdata()
{
 cout<<"Roll number"<<endl;
 cin>>n;
 cout<<"Name of student"<<endl;
 cin>>name;
 cout<<"Name of branch"<<endl;
 cin>>branch;
}
void student::getdata()
{
 cout<<"Roll number:"<<n<<endl;
 cout<<"Name of student:"<<name
                          <<endl;
```

Date : / /
Page No. :

```
    cout<<"Name of branch:"<<branch
                            <<endl;

}

int main()
{
  int n;
  string name;
  string branch;

  student o1;

  o1.setdata();
  o1.getdata();

  return 0;

}
```

Roll number
1
Name of student
Khush
Name of branch
CE

Roll number:
1
Name of student:
Khush
Name of branch:
CE

Page No. ____
__/__/__

17 Write a program to create a class for defining complex numbers and overload three set function. The first set function with takes no argument is used to create objects, second which takes one argument is used to initialize real and imaginary parts to equal values and third which takes two argument is used to initialized real and imaginary to two Different values. Define a display function that points the complex number.

```
#include <iostream>
 using namespace std;
 class complex
 {
   int r;
   int img;
   public:
          void setdata();
          void setdata (int a);
          void setdata (int a, int b);
          void display();

 };
```

```
void complex :: setdata()
{
    r = 1;
    img = 1;
}
void complex :: setdata(int a)
{
    r = a;
    img = a;
}
void complex :: setdata(int a, int b)
{
    r = a;
    img = b;
}
void complex :: display()
{
  cout<<r<<"+i"<<img<<endl;
}

int main()
{
  complex b1, b2, b3;

  b1.setdata();
  b2.setdata(3);
  b3.setdata(5, 6);

  b1.display();
```

```
b2.displayC();
b3.displayC();

return 0;

}
```

$1 + i1$

$3 + i3$

$5 + i6$

Page No. ____
__/__/__

Practical Set-5

18 Write a program to create class Test with one int member. Define constructor, destructor and getter for the same. Define a function Find square that takes object as an argument and returns square of int member of that object.

```
#include <iostream>
using namespace std;
class test
{
  int x;
  public:
      test(int x)
      {
        this -> x = x;
      }
      void setdata()
      {
        cout<<"x"<<x<<endl;
      }
      int getx()
      {
        return x;
      }
     ~test()
      {
        cout<<"Destructor is called"<<endl;
      }
};
```

```
int Find_square (test t1)
{
  int result;
  result = t1.getx() * t1.getx();
  return result;
}

int main()
{
  test t1(10);
  int ans;
  t1.setdata();
  ans = Find_square(t1);
  cout << "The square of x: " << ans << endl;
  return 0;
}
```

X 10
Destructor is called.
The square of X : 100
Destructor is called.

Page No. ____
__/__/__

19 Write a program to perform addition of two complex number using constructor overloading. Define add function outside the class that returns the addition. Also define ~~co~~ destructor.

```
#include <iostream>
using namespace std;
class complex
{

  int r, i;
  public:
      complex()
      complex(int r, int i)
      {
        this->r=r;
        this->i=i;
      }
      complex(int r)
      {
        this->r = this->i=r;
      }
      void display()
      {
        cout<<endl<<r<<"+j"<<i;
      }
```

```
~Complex()
{
    cout<<" Destructor is called"<<endl;
}

Friend complex C(complex &co1,
                 complex &co2);

};

complex C(complex &co1, complex &co2)
{
  complex ans;

  ans.r = co1.r + co2.r;
  ans.i = co1.i + co2.i;

  return ans;
}

int main()
{
  complex c1(2,3), c2(5,6), c3;
  c1.display();
  c2.display();
  c3=add C(c1,c2);

  c3.display();
  return 0;
}
```


mycompanion

$2 + 3i$

$5 + 6i$

$7 + 9i$

Page No. ____
__/__/__

20 Write a program to create class Time with members hours, minutes and seconds. Read value from keyboard and add two Time object by passing objects to function and display result. Also define destructor.

```
#include <iostream>
using namespace std;
class time
{
  int hour, min, sec;
  public:
        time()
        {
          hour = min = sec;
        }
        time(int h, int m, int s)
        {
          hour = h;
          min = m;
          sec = s;
        }
        void setdata()
        {
         cout<<"Enter the value"<<endl;
         cin >> hour >> min >> sec;
        }
```

```
    void getdata()
    {
     cout << hour << ":" << min << ":" << sec << endl;
    }
    ~time()
     {
       cout << "Destructor is called" << endl;
     }
};

int main()
{

  time t1, t2;

  t2.setdata();
  cout << "The time: " << endl;

  t1.getdata();
  t2.getdata();

  return 0;

}
```

Enter the value

23

24

25

The time

37:37:37

23:24:25

Destructor is called

Destructor is called.