2. Functions and Constructure

PAGE NO.:

Inline Functions :

If a function is to be define outside the class but still be created as a Internally defined function then such functions are clalled as Inline function.

Syntax :

```
inline Return_type Class_name :: Function name (argument list)
{
  Statement ;
}
```

Q write a program to find out area of circle using oop such that the class circle must have pre-Inline functions namely.
read() → To accept the radius
Compute() → To calculate the area.
display() → To display the result.

→
```
#include <iostream.h>
#include <conio.h>
Class Circle
{
Public :
int r;
Float a;
Void read();
Void Compute();
Void display();
};
```

```
inline Void circle :: read()
{
  cout<< "Enter r";
  cin >>r;
}
inline Void circle :: Compute()
{
   a = 3.14 * r*r;
}
inline Void circle :: display()
{
   cout<< "area of circle" << a;
}
Void main()
{
circle c;
c. read();
c. Compute();
c. display();
}
```

Q2. Write a program to find out Sum of following Series using Inline function.

$2^2 + 4^2 + 6^2 + 8^2 + \dots + n^2$

→

```
#include <iostream.h>
#include <conio.h>
Class Series {
Public :
 int n, i, sum;
```

```
Void read();
Void compute();
Void display();
};

Inline Void Series :: read()
{
Cout << "Enter the n";
Cin >> n;
}

Inline Void Series :: compute()
{
 Sum = 0;
for (i=2; i<=n; i++)
{
  Sum = Sum + (i*i)
}

}

Inline Void series :: display()
{
  Cout<<"Sum of Series"<<sum;

}

Void main()
{
   Series s;
   s.read();
   s.compute();
   s.display();
}
```

Q3. Write a program to accept two no. from user and Calculate addition, substraction, division & multiplication & display the result using inline function.

→

```
#include <iostream.h>
#include <conio.h>
Class Calculator
{ Public:
 int a,b, add, Sub, mul, div;
 Void read();
 Void addition();
 Void substraction();
 Void multiplication();
 Void division();
 Void display();
};

~~to~~ inline Void Calculator :: read()
{
 Cout << " Enter two Values";
 Cin >> a >> b;
}

inline Void Calculator :: addition()
{
  add = a+b;
}
inline Void Calculator :: Substraction()
{
  Sub = a-b;
}
```

```
inline void calculator :: multiplication()
{
  mul = a*b;
}
inline void calculator :: Division()
{
  div = a/b;
}
inline void calculator :: display()
{
  cout << "Addition :" << add;
  cout << "Substraction :" << sub;
  cout << "Multiplication :" << mul;
  cout << "Division :" << div;
}
void main()
{
  Calculator C;
  C.read();
  C.Addition();
  C.Substraction();
  C.Multiplication();
  C.division();
  C.display();
}
```

Constructor:

They are a special member function used for initializing the data elements of the object

⇒ Properties or charecterstics of constructor.

1] The name of constructor must always be same as that of class.

2] It has to be defined only in the public visibility of the class

3] It should not have any return type not even void.

4] The constructor is executed the movement an object of a class is created.

5] A constructor cannot be inherited but a derived class can call the constructor of the base class.

6] A constructor can not be static or virtual

7] Any number of constructor can be define in a class and such a set of constructor implements a concept called as Conistructor overloding.

PAGE NO.:

Types of Constructor

1] default Constructor
2] Parameterised Constructor
3] Copy Constructor.

1] Default Constructor:
If a constructor does not have any parameter list then it is called as default constructor.

Q. Write a program to find Area of circle using oop the value of the radius must be accepted from the user in the constructor. and the class circle must have two inline functions compute(), → Calculating area.
display() → displaying area.

→
```
#include <iostream.h>
#include <conio.h>
Class circle
{
Public:
 int r;
 Float a;
 circle()
{
 Cout<< " Enter r";
 Cin >>r;
}
 Void compute();
```

```
Void display();
};

inline Void circle :: compute()
{
  a = 3.14 * r * r;
}

inline Void circle :: display()
{
  Cout<< " Area of Circle:" << a;
}
Void main()
{
  circle c;
  c.compute();
  c.display();
}
```

Q. Write a program to calculate the value of the following series using default constructor and inline member function

$S = 1+2+3+\dots+n$;

→

```
# include <iostream.h>
# include <conio.h>
Class Series {
Public:
  int n, i, sum;
```

```
Series()
{
 cout<<" Enter n";
 cin>>n;
 }
Void compute();
Void display();
};

Void compute() Inline Void Series :: compute()
{
 Sum=0;
for (i=1; i<=n; i++)
 {
  Sum = Sum+i;
 }
}

Void displa Inline Void series :: display()
{
 cout<<" Sum of Series:" << Sum;
}

Void main()
{
 Series s;
 s. compute();
 s. display();
}
```

Q. Write a program to declare class major having data member add1, add2, add3 initialize the data members using constructor and store their addition in 3rd data member using function & display the addition.

→
```
#include <iostream.h>
# include <conio.h>
Class measure
{
 Public :
 int add1, add2, add3;
 measure()
 {
  add1 =100;
  add2 = 200;
 }
 Void addition()
 Void display();
};

Void measure :: addition ()
{
    add3 = add1 + add2;
}

Void measure :: display ()
{
  Cout<< "addition " << add3;
}
```

```
Void main()
{
 measure m;
 m.addition();
 m.display();
}
```

Q. Write a program to calculate factorial of a number find out value of a no from user using constructor and in class factorial use two Externally defined member-function.

→
```
#include <iostream.h>
#include <conio.h>
Class Factorial
{
Public:
int i, n, ~~mul~~ fact;
Factorial()
{
 cout << " Enter n";
 cin >> n;
}
 Void compute();
 Void display();
};

Void factorial :: compute()
{
 fact = 1;
 for (i=1 ; i<=n ; i++)
```

```
{
fact = fact * i;
}
}
Void Factorial :: display()
{
cout << " Factorial of a number: " << fact;
}
Void main()
{
factorial f;
f. Compute();
f. display();
}
```

2] Parameterised Constructor:
If a constructor does have a parameter list that is which accept one or more parameters then it is called as Parameterised Constructor.

Syntax:

Class_name object_name (argument list);

Q. Write a program to find out area of circle using oop the value of the radius must be accepted from the user in the main program and passed to the parameterised constructor. and the class circle must have two inline function namely compute() → to calculate area display() → to displaying the reason.

→
```
#include <iostream.h>
#include <conio.h>
Class circle
{
Public:
 int a, r;
 circle (float x)
 {
  r = x;
 }
 Void compute();
 Void display();
};
```

```
inline Void circle :: compute()
{
a = 3.14 * r * r;
}
inline Void circle :: display()
{
cout << " Area : " << a;
}
Void main()
{
cout << "Enter r";
cin >> r;
circle c(r);
c.compute();
c.display();
}
```

Q. Write the program to calculate the value of the following Series using Parameterised Constructor and inline member function. $S = 2^2 + 4^2 + 6^2 + \dots + n^2$

→
```
#include <iostream.h>
#include <conio.h>
Class Series
{ Public:
 int i, n, sum ;
 Series (int x)
 {
 n = x;
 Cout << " Enter x";
 Cin >> x;
}
```

```
Void compute();
Void display();
};

inline Void Series :: compute()
{
  Sum = 0;
for (i=2; i<=n ; i++)
{
for (i=2; i<=n; i+=2)
{
  Sum = Sum + (i*i);
}
}

inline Void Series :: display()
{
Cout << " Sum of series :" << Sum;
}

Void main()
{
Series s (x);
S. compute();
S. display();
}
```

Q. Write a program to calculate sum of all odd numbers between 1 to n using parameterised constructor & inline member function.

→
```
#include <iostream.h>
#include <conio.h>
Class Series
{
Public :
int n, i, Sum ;
Series (int x)
{
 n = x ;
}
Void compute();
Void display();
};
Void Compute
inline Void Series :: Compute()
{
 Sum = 0;
 For (i=1 ; i<=n ; i++)
{
 if (i%2 == 1)
{
 Sum = Sum + i;
}
}
}
```

```
inline void series :: display()
{
cout << " sum of odd no: " << sum;
}

void main()
{
 cout << "Enter x";
 cin >> x;
series S(x);
S. compute();
S. display();
}
```

3] Copy Constructor :
IF a constructor has an object of the same class as a parameter in the argument list that is called as copy constructor.

The Copy constructor uses an object of the same class to initialize another object of a class.

Q) Write a program to find area of circle using oop the value of the radius must be accepted from the user in the main program and passed to the copy constructor and the class circle must have two inline functions namely
Compute → To calculate area
display → For displaying the result

→
```
#include <iostream.h>
#include <conio.h>
Class circle
{
Public :
int a,r ;
circle (float x)
{
 r =x;
}
circle (circle & c)
{
   r = c.r;
}
```

```
Void Compute();
Void display();
};

inline Void circle :: Compute()
{
a = 3.14*r*r;
}
inline Void circle :: display()
{
cout << "area = " << a;
}

Void main()
{
cout << "Enter radius";
cin >> r;
circle C(r);
C.Compute();
C.display();
circle C1(C);
C1.Compute();
C1.display();
}
```

Q2. Write a program to calculate the value of the following series using copy constructor & inline member function. $1^2 + 2^2 + 3^2 + \_\_ + n^2$

→

```
#include <iostream.h>
#include <conio.h>
Class Series
{
 Public :
 int i,n, Sum ;
  Series ( int x)
 {
    n = x ;

 }

 Series (Series &s)
 {
    ~~n = n.x~~
  n = s.n;
 }
 Void compute();
 Void display();
 };

 inline Void Series :: Compute()
 {
   Sum = 0;
  for (i=1 ; i<=n ; i++)
  {
   Sum = Sum+(i*i);
  }
 }
```

```
inline void series :: display()
{
cout<<"Sum of series" <<sum;
}

void main()
{
cout<<"Enter no";
cin >>x;
Series S(x);
S. compute();
S. display();
Series S1 (S);
S1. compute();
S1. display();

}
```

Q. Write a program to calculate factorial of a number using copy constructor inline member function.

→

```
#include <iostream.h>
#include <conio.h>
Class Factorial
{
public :
int i,n, fact;
Factorial (int x)
{
  n=x;
}
```

```
Factorial (Factorial &F)
{
    n = F.n;
}
Void compute();
Void display ();
};

inline Void Factorial :: compute()
{
 Fact = 1;
for (i=1; i<=n; i++)
{
 Fact = Fact * i;
 }
}
inline Void Factorial :: display()
{
 Cout<< " Factorial :" << Fact;
}
Void main()
{
 Factorial
 Cout<<" Enter number";
  Cin >> x;
  Factorial F(x);
  F. compute();
  F. display();
```

```
Factorial F1(f);
F1.compute();
F1.display();
}
```

* Destructor:

A destructor is used to destroy the memory Allocated by the constructor

Carecteristics of destructor.

- It is defined only in the public visiblity of class
- The name of the destructor must be same as that of class prefix with a tilde sign (~)
- It canot return or accept any value it is automaticly called when the compiler exits from the main programm
- The destructor can only destroy the memory location allocated by the constructor and the constructor can allocate memory only using the new operator.

- To destroy this memory allocated by the new operator we require delete operator in this case the delete operator must be in the destructor.

```
#include <iostream.h>
#include <conio.h>
class test
{
 public:
 int *p;
 test()
 {
  p = new int;
 }
 void read()
 {
  cout << "Enter Value";
  cin >> *p;
 }
 void display()
 {
  cout << "Value : " << *p;
 }
 ~test()
 {
  delete p;
  cout << "Destroyed";
 }
};
```

```
Void : main()
{
test t;
t.read();
t.display();
}
```

Difference between Constructor & Destructor

Constructor	Destructor
① A Constructor is a special member function whose task is to insalize the object of its class	① A Destructor is a special member function whose task is to destroy the object of that have been created by the Constructor.
⑪ Constructor are classified in various type such as ① Default Constructor ② parameterised ③ Copy Constructor ④ overloaded Constructor or multiple Constructor.	⑪ Destructors are not classified in any type
⑪ It is invoke automaticly when the objects are created	⑪ It is invoke implicitely by the compiler upon exit of Program.

(iv) Constructor accept parameters also it can have default value for its parameter	(iv) Destructor never accept parameters
(v) A class can have more than one constructors	(v) A class can have at the most one destructor
(vi) It construct the values of data member of a class	(vi) It does not construct the values for the data member of the class

(vii) Syntax:
Class name()
{
};
	(vii) syntax:
	~Classname()
	{
	};