



AISSMS

COLLEGE OF ENGINEERING

ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade



DEPARTMENT OF COMPUTER ENGINEERING

UNIT TEST I

SET A

Class: SE (Shift-II)

Subject: OOP

Date: 7/07/2019

Q.1	Describe member access control with suitable examples.	[05]	CO1
Q.2	How to create static data members.	[05]	CO1
Q.3	Elaborate term inline function with suitable example	[05]	CO1

S. S. Jodhar

I. K. D.
H.O.D.
Computer Engg Dept
AISSMS COE Pune



PRINCIPAL
ALL INDIA SHRI SHIMAJI MEMORIAL SOCIETY'S
COLLEGE OF ENGINEERING
KENNEDY ROAD, PUNE-411 001

Q.1	<table><tr><th colspan="2">Member-Access Control</th></tr><tr><th>Type of Access</th><th>Meaning</th></tr><tr><td><u>private</u></td><td>Class members declared as private can be used only by member functions and friends (classes or functions) of the class.</td></tr><tr><td><u>protected</u></td><td>Class members declared as protected can be used by member functions and friends (classes or functions) of the class. Additionally, they can be used by classes derived from the class.</td></tr><tr><td><u>public</u></td><td>Class members declared as public can be used by any function.</td></tr></table>	Member-Access Control		Type of Access	Meaning	<u>private</u>	Class members declared as private can be used only by member functions and friends (classes or functions) of the class.	<u>protected</u>	Class members declared as protected can be used by member functions and friends (classes or functions) of the class. Additionally, they can be used by classes derived from the class.	<u>public</u>	Class members declared as public can be used by any function.	[05]	CO1
Member-Access Control													
Type of Access	Meaning												
<u>private</u>	Class members declared as private can be used only by member functions and friends (classes or functions) of the class.												
<u>protected</u>	Class members declared as protected can be used by member functions and friends (classes or functions) of the class. Additionally, they can be used by classes derived from the class.												
<u>public</u>	Class members declared as public can be used by any function.												
Q.2	We can define class members static using static keyword. When we declare a member of a class as static it means no matter how many objects of the class are created, there is only one copy of the static member. <pre>#include <iostream> using namespace std; class Box { public: static int objectCount; // Constructor definition Box(double l = 2.0, double b = 2.0, double h = 2.0) { cout << "Constructor called." << endl; length = l; breadth = b; height = h; // Increase every time object is created objectCount++; } double Volume() { return length * breadth * height; } private: double length; // Length of a box double breadth; // Breadth of a box double height; // Height of a box }; // Initialize static member of class Box int Box::objectCount = 0; int main(void) { Box Box1(3.3, 1.2, 1.5); // Declare box1 Box Box2(8.5, 6.0, 2.0); // Declare box2 // Print total number of objects. cout << "Total objects: " << Box::objectCount << endl; return 0; }</pre>	[05]	CO1										
Q.3	C++ inline function is powerful concept that is commonly used with classes. If a function is inline, the compiler places a copy of the code of that function at each point where the function is called at compile time. Any change to an inline function could require all clients of the function to be recompiled because compiler would need to replace all the code once again otherwise it will continue with old functionality. To inline a function, place the keyword inline before the function name and define the function before any calls are made to the function. The compiler can ignore the inline qualifier in case defined function is more than a line. A function definition in a class definition is an inline function definition, even without the use of the inline specifier.	[05]	CO1										

S.S. Jadhav

H.O.D.
Computer Engg Dept.
AISSMS COE Pune



AISSMS

COLLEGE OF ENGINEERING

ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade



DEPARTMENT OF COMPUTER ENGINEERING

UNIT TEST I

SET B

Class: SE (Shift-II)

Subject: OOP

Date: 17/07/2019

Q.1	What is static member function? Give example for the same.	[5]	CO1
Q.2	Write a short note on reference variable with example.	[5]	CO1
Q.3	How to create constructor and destructor in c++.	[5]	CO1

Sodher
S.S. Sodher

Vinod
H.O.D.
Computer Engg Dept
AISSMS COE Pune



[Signature]
PRINCIPAL
ALL INDIA SHRI SHIMAJI MEMORIAL SOCIETY'S
COLLEGE OF ENGINEERING
KENNEDY ROAD, PUNE-411 001

Q.1	A static member function is a special member function, which is used to access only static data members, any other normal data member cannot be accessed through static member function. Just like static data member, static member function is also a class function; it is not associated with any class object. We can access a static member function with class name, by using following syntax: <pre>class_name:: function_name(parameter);</pre> <pre>#include <iostream> using namespace std; class Demo { private: static int X; static int Y; public: static void Print() { cout << "Value of X: " << X << endl; cout << "Value of Y: " << Y << endl; } }; int Demo :: X =10; int Demo :: Y =20; int main() { Demo OB; cout<<"Printing through object name:"<<endl; OB.Print(); cout<<"Printing through class name:"<<endl; Demo::Print(); return 0; }</pre>	[5]	CO1
Q.2	A reference variable is an alias, that is, another name for an already existing variable. Once a reference is initialized with a variable, either the variable name or the reference name may be used to refer to the variable. <pre>#include <iostream> using namespace std; int main () { int i; double d; int& r = i; double& s = d; i = 5; cout << "Value of i : " << i << endl; cout << "Value of i reference : " << r << endl; d = 11.7; cout << "Value of d : " << d << endl; cout << "Value of d reference : " << s << endl; return 0; }</pre>	[5]	CO1
Q.3	Constructors are special class functions which performs initialization of every object. The Compiler calls the Constructor whenever an object is created. Constructors initialize values to object members after storage is allocated to the object. Whereas, Destructor on the other hand is used to destroy the class object. Constructor Syntax <pre>class A { public: int x; // constructor A() { // object initialization } };</pre> Destructor Syntax <pre>class A { public: // defining destructor for class ~A() { // statement } };</pre>	[5]	CO1

Adhar
S.S. Adhar

H.O.D.
H.O.D.
Computer Engg Dept
AISSMS COE PUNE



ALL INDIA SHRI SHIVAJI MEMORIAL SOCIETY'S
COLLEGE OF ENGINEERING

KENNEDY ROAD, PUNE - 411 001.

Supervisor's Signature

[Signature]

Name: Tejas Dhah Roll No.: 18CS050
Subject: OOP Division: Comp II
Examination: UNIT-I Day & Date: 17/7/19

Question No.	1	2	3	4	5	6	7	8	9	10			Total Marks
Marks	4	5	4	—	—	—	—	—	—	—			13/15

Examiner Signature

[Signature] each (5mk)

QI] what is static member function? give

eg

QII] Write a short note on reference variable

QIII] How to create Constructor and destructor

→

QIII] a) Constructor and destructor. Contains same name that of class.

b) Constructor - It is a function which gets automatically called when ~~function~~ class object is defined

Syntax - class class_name

```

{
    class_name() → Constructor
    {
        ....
    }
}

```

Constructor does not have a return type



PRINCIPAL
ALL INDIA SHRI SHIVAJI MEMORIAL SOCIETY'S
COLLEGE OF ENGINEERING
KENNEDY ROAD, PUNE-411 001

III

Destructor - It is used to ~~destroy~~ destroy the object created (ie delete).

It is a good programming technique as ~~it~~ it deletes the object which created the space.

eg

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

```
Class Conde
```

```
{
    Conde()
    {
        cout << "This is the constructor";
    }

```

```
~Conde()
```

```
{ cout << "The object is deleted";
}
```

```
int fun() { cout << "This is the function";
}
```

```
int main()
```

```
{
    Conde a; a.fun();
    return 0;
}
```

Output

This is the constructor The object is deleted



Q II

Reference variable -

(a) A variable ~~which~~ which addresses another variable is called reference variable.

(b) It is used to declare globally (ie its scope is throughout program)

Syntax -

$\&abc = xyz$

here abc is reference.

eg -

```
#include <iostream>
```

```
using namespace std;
```

```
int main ()
```

```
{ int xyz, abc;
```

```
xyz = 100;
```

```
&abc = xyz;
```

```
cout << "The value of xyz variable = " << xyz;
```

```
cout << "The value of   
variable reference = " << abc;
```

```
xyz = 150;
```

```
cout << "The change value of variable = " << xyz;
```

```
cout << "Now the of reference variable = " << abc;
```

```
}
```



Output :

The value of variable = 100

The value of reference variable = 100

The Changed Value of variable = 150

Now the value of reference variable = 150

Q1] Static ~~mem~~ member function.

- ① Static member function is also a function in which only one copy is created while ~~and~~ an object of a class is declared.
- ② Static member function implies to a class.
- ③ ~~In~~ In static ~~mem~~ member we can declare any variables;

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

```
Class static_member
```

```
{ public int a; public: int a = 0;
  static member ()
```

```
{
  cout << "This is a constructor";
```

```
static int fun ()
```

```
{
  cout << "This is a static member" << a;
  a++ }
```

```
};
```



~~int main ()~~

~~{~~

~~B b;~~

~~return 0~~

~~}~~

~~O/P.~~

~~'B' is initialized.~~

~~only~~

int main

{

static_member a; a.fun();

static_member b; b.fun();

}

O/P.

(4)

This is Constructor

This is a static member 0

This is a Constructor

This is a static member 1



PRINCIPAL
ALL INDIA SHRI SHIVAJI MEMORIAL SOCIETY'S
COLLEGE OF ENGINEERING
KENNEDY ROAD, PUNE-411 001