Name : Patel Rudra Ketan

ID : 22CEUOS135

Roll no. : CE100

Batch : B1

Lab09(sessional 2 paper)

Q.1 (A)Choose the most appropriate alternate(s).
(a) Which of the following can have return type?
(a) constructor (b) destructor (c) copy constructor (d) none of the above
(b) Which of the following forms of inheritance can create multiple copies of the base class in the derived class object?
(a) hybrid inheritance (b) hierarchical inheritance
(c) multiple inheritance (d) multilevel inheritance
(c) A derived class can
(a) extend the functionalities of the base class
(b) override the features of the base class

(c) use the attributes of the base class
(d) all of the above
(d) What type of members of the base class are accessible by the members of a derived class for the public inheritance of the base class?
(a) private only (b) public only
(c) public and protected (d) protected only
(e) Protected members can be
(a) accessed from the main(). (b) inherited and

accessed by a derived class.
(c) accessed from another class. (d) all of the above.
(f) friend functions and classes can access
(a) private, protected and public members of the class
(b) private and protected members of the class
(c) private and public members of the class
(d) protected and public members of the class
(g) Which of the following cannot be a correct form of invoking base class constructors?

(a) D(int a1, int a2, float b1, float b2, int d1): A(a1,a2), B(a1,b1,b2)
(b) D(int a1, int a2, float b1, float b2, int d1): A(a1,a2), B(b1,b2,c1)
(c) D(int a1, int a2, float b1, float b2, int d1): A(a1,a2), B(a1,a2)
(d) D(int a1, int a2, float b1, float b2, int d1): A(a1,a2,b1), B(b1,b2)
(h) What will be order of execution of construction for the following declaration? class A: public B, virtual public C, virtual D
(a) C( ),D( ),A( ),B( ) (b)A( ),B( ),C( ),D( )
(c) C( ),D( ),B( ),A( ) (d) D( ), C( ), B( ), A( )
Q.1 (B) Do as Directed.
(a) State true/false with reason(s).
(i) The only way an object can be passed to a constructor of the same class is a copy constructor.
=> Yes ,it is the only way an object`s refrence can be passed through constructor.
(ii) Constructors and destructors for the objects of a

program are executed in the same order.
=> No, destructor will executed in reverse order.
(b) Identify and correct the syntax and logical error(s) in class definition only in the following code to get output as 8.

```
#include <iostream>
using namespace std;
class Student{
public:
intRollNumber;
void PrintDetails() const
{cout<<RollNumber<< "\n"; }
};
int main()
{
Student Student1;
Student1.RollNumber = 8;

const Student OtherStudent = Student1;
OtherStudent.PrintDetails();
}
```

non-const function can`t called by const members or object.

Q.2 Attempt Any TWO from the following questions.
(a) Write the code of following methods for the vector class:
Note: Vector size must remain same after performing following tasks.
(i) Define a vector V of type integer with a size of 8.
(ii) Set the first element to 10.
(iii) Set the last element to 18.

(iv) Display the elements of vector.

=>

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

int main()
{
vector<int> V(8,0);
V.front()=10;
V.back()=18;
for(int i=0;i<8;i++)
cout<<V[i]<<" ";
return 0;
}
```

(b) (i)Define an appropriate class and function definitions for following main( ) and predict the output of the code.

=>

```
#include "iostream"
using namespace std;
class Area{
int len;
public:
void SetArea(int x){

len=x;
}
void ShowArea(){
cout<<len<<endl;
}
void SqLength(){
len*=len;
```

```
}
friend Area Sq(Area &);
};
Area Sq(Area &x){
Area tem;
tem.len = x.len * x.len;
return tem;
}
int main(){
Area Area1, Area2, Area3;
Area1.SetArea(12);

Area1.ShowArea();
Area2 = Area1;
Area2. SqLength();
Area2.ShowArea();
Area3 = Sq(Area1);
Area3.ShowArea();
return 0;}
```

A (ii)Identify and correct the syntax and logical error(s) in following code and write the output after correction(s).

```
#include <iostream>
using namespace std;
class Student {
public:
static int TotalStudents=0; };
int Student::TotalStudent;
int main( )
{ cout <<Student::TotalStudents << "\n";

return 0;
```

}
OUTPUT: 0
CO4 R (c) What is inheritance? Explain the types of inheritance in brief.
=> Inheritanceis a mechanism of driving a new class from an existing class. The existing (old) class is known as base classor super classor parent class. The new class is known as a derived classor sub classor child class. It allows us to use the properties and behavior of one class (parent) in another class (child).
Single Inheritance
In single inheritance, a sub-class is derived from only one super class. It inherits the properties and behavior of a single-parent class. Sometimes it is also known as simple inheritance.
Multi-level Inheritance
A class is derived from a class which is also derived from another class is called multi-level inheritance. In simple words, we can say that a class that has more than one parent class is called multi-level

inheritance. Note that the classes must be at different levels. Hence, there exists a single base class and single derived class but multiple intermediate base classes.
Hierarchical Inheritance
If a number of classes are derived from a single base class, it is called hierarchical inheritance.
Multiple Inheritance
A single derived class inheriteted several base class,it called the multiple inheritance.

Q.3 Answer the followings Questions.
CO4 N (a) (i)Find the missing declaration or statements and correct the incorrect statements for proper operation of the program given below.
(ii) Also predict the output of the complete program with corrected statements.
(iii)What will happen if show( ) of the derived class is removed from the class definition?
Justify your answer.
(iv)What will be the size of the object of the derived

class? How?

```
#include <iostream>
using namespace std;
class base1
{
int i;
public:
base1(){cout << "Constructing base1 without
argument\n ";}
base1(int x){i = x;
cout << "Constructing base1 with argument\n";}
~base1() { cout << "Destructing base1\n"; }
void show() { cout << i << " "; }
};
class base2
{
char c;
public:

base2()
{
```

```
cout << "Constructing base2 without argument\n";
}
base2(char x){
cout<<"Constructing base2 with argument\n";
c=x;
}
~base2() { cout << "Destructing base2\n"; }
void show() { cout << c << " "; }
};
class derived : public base1, public base2
{
int j;
public:
derived(int x, int y, char p): base2(p),base1(y){
j = x;
cout << "Constructing derived with argument\n ";

}
~derived(){cout << "Destructing derived\n "; }
void show(){
base1::show();
base2::show();
cout<<j<<" "<<endl;
}
};
int main()
{
derived ob(3, 4, 'A');
ob.show();
return 0;
}
```

OUTPUT:

Constructing base1 with argument
Constructing base2 with argument
Constructing derived with argument

4 A 3
Destructing derived
Destructing base2
Destructing base1
CO3 U (b) List any two situations in which a copy constructor is used?
1) when explicit construction will called then the copy construction called.
Like complex c1= complex (c2);
complex c3(c2);
2) copy constructor will also called when the function argument copy and their return object of a
class.
CO3 N (a) (i) Predict the output of the following program.
(ii) What is missing in the program for its appropriate behaviour?
(iii) Insert the appropriate feature in the program for the desired behaviour and show the change

in the predicted output.
(iv)How many times the destructor of the class array will be executed? Write a suitable
destructor for the class array. Why a user defined destructor is desirable for this class?
OUTPUT:
9
8

7
6
5
4
3
2
1
0
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10

MISSING BEHAVIOUR : copy constructor was missing.

```
#include <iostream>
using namespace std;
class list
{
int *t;
public:
list()
{
t = new int[10];
for (int i = 0; i < 10; i++)
t[i] = i;
}
```

```
void put(int i, int j)
{
if (i >= 0 && i < 10)
t[i] = j;
}
int get(int i) { return t[i]; }
list (const list &old){
t = new int[10];
for (int i = 0; i < 10; i++)
t[i] = old.t[i];
}
~list(){
cout<<"destructor called"<<endl;
delete[] t;
}
};
int main()
{
list beta;
int i;
for (i = 9; i >= 0; i--)
cout << beta.get(i) << "\n";
list alpha(beta);
for (i = 0; i < 10; i++)
beta.put(i, i + 1);
for (i = 0; i < 10; i++)
cout << beta.get(i) << " " << alpha.get(i)
<< endl;
return 0;
}
```

Two time destructor was called..

CO4 U (b) Briefly explain function overriding with reference to inheritance.

=>Function overriding provides you with a way to override an existing functionality of a class inside a particular derived class. This can be useful when a child class requires its own version of a functionality. Like base class have one function of name ‘print()’ and derived class have also same name function then , derived class overriding base function.