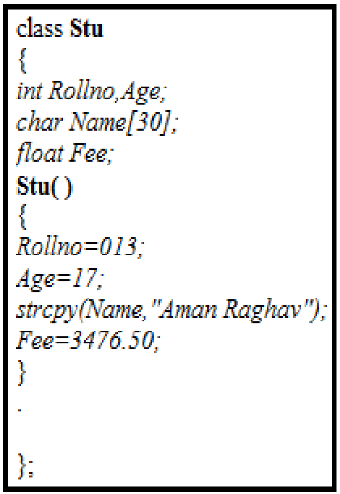
CONSTRUCTORS AND DISTRUCTORE

(*Create memory and Free Memory Allocat and Dallocat*)

Constructor is used to initialize initial value to Data Member of a ***class****A member function with the same as its class is called Constructor and it is used to initialize theobject of that class with a legal initial value.*

Important Points (topics) with example:

1. No return type
2. Work like a function
3. Name Same as class name
4. Default and parameterize like function
5. Only **default constructor** Call automatically when we create object. If constructor is parameterize it called by passing value with object
6. Should be in public section
7. **If a class have some DM or MF and parameterize constructor it should be have default constructor**



```
class Stu
{
int Rollno,Age;
char Name[30];
float Fee;
Stu( )
{
Rollno=013;
Age=17;
strcpy(Name,"Aman Raghav");
Fee=3476.50;
}
.

};
```

Play with C+

8. 1. What is Constructor?

It is a member function which is automatically used to initialize the objects of the class type with legal initial values.

2 Why you should define a Constructor ?

Uninitialized member fields have garbage in them. This creates the possibility of a serious bug (eg, an uninitialized pointer, illegal values, inconsistent values, ...).

3 How can we declaring a constructor.

A constructor is similar to a function, but with the following differences.

- No return type.
- No return statement.

4. Define any five characteristics of Constructor.

- They are called automatically when the objects are created.
- All objects of the class having a constructor are initialize before some use.
- The address of a constructor cannot be taken.
- These cannot be static.
- Return type cannot be specified for constructors.

```
class Point
{
public:
Point();                          // parameterless default constructor
Point(int new_x, int new_y);      // constructor with parameters
int getX();
int getY();
private:
int x;
int y;
};
```

5 What are the types of Constructor (Write any Four.)?

- Default constructors
- Parameterized Constr6uctor

6 What is Default Constructor?

A default constructor is a constructor that either has no parameters, or if it has parameters, all the parameters have default values.

7 What is Parameterized Constructor?

A Constructor with arguments is called a parameterized constructor.

8 What is Overloaded Constructor?

Like function Overloading Constructor Overloading can also be done.

9 What is Copy Constructor? Explain with example.

Copy constructor is

- a constructor function with the same name as the class
- used to make deep copy of objects.

For ex:

```
class A //Without copy constructor
{
private:
int x;
public:
A() {A = 10;}
~A() {}
}
```

10 Explain any three important places where a copy constructor is called.

- When an object is created from another object of the same type
- When an object is passed by value as a parameter to a function
- When an object is returned from a function

11 What is Dynamic Initialization of objects?

If we initialized class objects at run time, it is the case of Dynamic Initialization.

12 Define Destructors. With Syntax.

Destructors are less complicated than constructors. You don't call them explicitly (they are called automatically for you), and there's only one destructor for each object. The name of the destructor is the name of the class, preceeded by a tilde (~). Here's an example of a destructor:

```
Player::~Player() {
strength = 0;
agility = 0;
health = 0;
}
```

13 What are the characteristics of Destructors? Any Five

- These are called automatically when the objects are destroyed.
- Destructor functions follow the usual access rules as other member functions.
- No arguments and return type permitted with destructors.
- These cannot be inherited.
- Address of a destructor cannot be takenExample :

```
class Student
{
int rollno;
float marks;
public:
student( ) //Constructor
{
rollno=0;
marks=0.0;
}
//other public members
};
```

TYPES OF CONSRUCTORS:

1. Default Constructor:

A constructor that accepts no parameter is called the Default Constructor. If you don't declare a constructor or a destructor, the compiler makes one for you. The default constructor and destructor take no arguments and do nothing.

2. Parameterized Constructors:

A constructor that accepts parameters for its invocation is known as parameterized Constructors , also called as Regular Constructors.

DESTRUCTORS:

- A destructor is also a member function whose name is the same as the class name but is preceded by tilde("~").It is automatically by the compiler when an object is destroyed. Destructors are usually used to deallocate memory and do other cleanup for a class object and its class members when the object is destroyed.
- A destructor is called for a class object when that object passes out of scope or is explicitly deleted.

Example :

```
class TEST
{ int Regno,Max,Min,Score;
Public:
TEST( ) // Default Constructor
{ }
TEST (int Pregno,int Pscore) // Parameterized
Constructor
{
Regno = Pregno
;Max=100;Max=100;Min=40;Score=Pscore;
}
~ TEST ( ) // Destructor
{ Cout<<"TEST Over"<<endl;}
};
```

The following points apply to constructors and destructors:

- Constructors and destructors do not have return type, not even void nor can they returnvalues.
- References and pointers cannot be used on constructors and destructors because theiraddresses cannot be taken.
- Constructors cannot be declared with the keyword virtual.
- Constructors and destructors cannot be declared static, const, or volatile.
- Unions cannot contain class objects that have constructors or destructors.
- The compiler automatically calls constructors when defining class objects and calls destructors when class objects go out of scope.
- Derived classes do not inherit constructors or destructors from their base classes, but they do call the constructor and destructor of base classes.
- The default destructor calls the destructors of the base class and members of the derivedclass.
- The destructors of base classes and members are called in the reverse order of thecompletion of their constructor:
- The destructor for a class object is called before destructors for members and bases arecalled.

Copy Constructor

Constructor & destructor*: A class constructor is a special function in a class that is called when a new object of the class is created.*

A destructor: is also a special function which is called when created object is deleted.

C++ copy constructor: The copy constructor is a constructor which creates an object by initializing it with an object of the same class, which has been created previously.

- **A copy constructor is a special constructor in the C++ programming language used to create a new object as a copy of an existing object**.

Play with C++

Play with C++

- A copy constructor is a constructor of the form **classname(classname &).**The compiler will use the copy constructors whenever you initialize an instance using values of another instance of the same type.
- Copying of objects is achieved by the use of a copy constructor and a assignment operator.

Example :

```
class Sample{ int i, j;}
public:
Sample(int a, int b) // constructor
{ i=a;j=b;}
Sample (Sample & s) //copy constructor
{ j=s.j ; i=s.j;
Cout <<"\n Copy constructor working \n";}
void print (void)
{cout <<i<< j<< "\n";}
```

Note : ***The argument to a copy constructor is passed by reference, the reason being that when an argument is passed by value, a copy of it is constructed. But the copy constructor is creating a copy of the object for itself, thus ,it calls itself. Again the called copy constructor requires another copy so again it is called.in fact it calls itself again and again until the compiler runs out of the memory .so, in the copy constructor, the argument must be passed by reference.***

The following cases may result in a call to a copy constructor:

- **When an object is passed by value to a function**: The pass by value method requires a copy of the passed argument to be created for the function to operate upon .Thus to create the copy of the passed object, copy constructor is invoked

If a function with the following prototype :

void cpyfunc(Sample); // Sample is a class

then for the following function call

cpyfunc(obj1); // obj1 is an object of Sample type

the copy constructor would be invoked to create a copy of the obj1 object for use by cpyfunc().• **When a function returns an object** : When an object is returned by a function the copy constructor is invoked

Sample cpyfunc(); // Sample is a class and it is return type of cpyfunc()

If func cpyfunc() is called by the following statement

obj2 = cpyfunc();

Then the copy constructor would be invoked to create a copy of the value returned by cpyfunc() and its value would be assigned to obj2. The copy constructor creates a temporary object to hold the return value of a function returning an object.

Important Points (topics) with example:

1. No return type

2. Work like a function

5. Only **default constructor** Call automatically when we create object. If constructor is parameterize it called by passing value with object

3. Name Same as class name

4. Default and parameterize like function

6. Should be in public section

7. **If a class have some DM or MF and parameterize constructor it should be have default constructor**

9. **Declare**

class **class-name**
{
public:
DM
MF
class-name()
{
//default
}
class-name(Arguments)
{
// parameterize Const.
}
} ;
void main()
{
//Create objects………}
	Case 1(Wrong or Right)
	class **Student**
	{
	public:
	int **a=6**;//error Not allow to initialize
	int **b=7**;// error Not allow to initialize
	};
	void main()
	{
	Student x={6,7}; // default Value
	Or
	Student x;
	x.a=6;// default Value of a
	x.b=7;// default Value of b
	or
	cin>>x.a>>x.b;//Dynamic }
		Case 2 Right
		class **Student**
		{
		public: int a,b;
		Student()
		{
		a=6; b=7;
		cout<< "Default const";
		cout<<a<":"<<<b;}
		Student(int a)
		{
		**cout<< "Parameterize constructor"<<a;}**};
		void main()
		{
		Student x; // default call
		Student y(4); } //parameterize call

Invalid or Valid case of a constructor calling:

Case 1(Wrong)	**Case 2(Wrong)**	**Case 3(Wrong)**	**Case 4 (Right)**

class **A**
{
A()
{
cout<< “Default Con”;
}
};
void main()
{
A x; **//Invalid Private Con**
	class **A**
	{
	public:
	A(int a,int b)
	{
	cout<< “ Parameterize Con”;
	}
	};
	void main()
	{
	A x;
	//Invalid Parameterize Con A y(10,20)
	}
		class **A**
		{
		public: int a1;
		void dis()
		{
		cout<< “Member Function”;
		}
		A(int a, int b)
		{
		cout<< “ Para Cons”;
		}
		};
		void main()
		{**A y(10,20)**
		// Parameterize Const
		y.a1=10; //Invalid
		y.dis(); //Invalid
		}
			class A
			{
			public: int a1;
			void dis(){
			cout<< “Member Function”;
			}
			A(){
			cout<< “Default Constructor”;}
			A(int a, int b)
			{**cout<< “ Parameterize Cons”;**
			}
			};
			void main()
			{
			A x; // Default constructor
			A y(10,20);// Parameterize Const.
			x.a1=10;//valid
			x.dis();//valid
			}

10. **Copy Constructor:** A technique to Initialize value of one object to another object

```
A x(3,5);// Assigned/initialize value to x
A y=x;     // Assigned/copied value of x to y
A y(x);    // Assigned/copied value of x to y
```

Example

```
Class A
{
public :
int a,b;
A(int a1,int b1)// Constructor 1
{
a=a1;
b=b1;
cout << “Constructor 1”<<a<< “:”<<b;
 }
A( A &z)          // Copy Constructor
{
a=z.a*2;
b=z.b*3;
cout << “Copy Constructor ”<<<a<< “:”<<b;
}
};
void main()
{
A x(3,5);
A y=x;
A y(x);
}
```

Output :

```
Constructor  1          3:5
Constructor  1         3:5
Copy Constructor      6:15
```

Declaration, Definition and calling of Constructor :

A(); //Declaration
A() // Definition
{
::
}
A x; //Calling

11. A constructor arguments with default value work like default or parameterize :

Declaration :

```
A(int a=7,int b=73)
{
…
.
.
.
}
```

Calling :

```
A x;
A y(10);
A z(10,20); //all 3 objects are working….
```

12. Constructor overload :

Same name as class but different arguments:

```
class A
{
public
A();                          //default
A(int a);                     //parameterize 1 integer
A(int a,int b);               //parameterize 2  integer
A(char name[],int age); //parameterize name and age
};
```

Play with C++

13. Constructor as outside:

```
class A
{
public
A(int a);
};
A::A(int a)
{
:::::::::::::}
```

14. Assign value to data member of class on the time of defining constructor:

Case 1

```
class A{
 int a,b;
public:
A(int i, int j)
{
  a=i;
 b=j;
cout<< a<< " :"<<b;}
};
void main()
{
A x(3,7);
}
```

Case 2

```
class A
{
int a,b;
public:
A(int i, int j) :a(i),b(j)
{
cout<< a<< " :"<<b;
}
};
void main()
{
A x(3,7);
}
```

15. Other method of calling Parameterize Constructor:

1. **A x(3,4);**
2. **A x=A(3,4);**

16. Destructors: it is used to free the memory space which allocated by Constructor

1. Like a constructor but should be default **(no argument)**
2. With negation symbol **(Tilde) ~**
3. In public section
4. Only one destructor in one class
5. Call automatic after execution of all constructor of **a class or program**in **reverse order.**

Example:

```
class A
{
public :
 A() // Constructor Definition
{
cout<< "Constructor";
}
~A() // Destructor Definition
{
cout<< "Destructor";
}
};
```

```
void main()
{
A x,y,z;
}
```

Output:

```
Constructor
Constructor
Constructor
Destructor
Destructor
Destructor
```

Solve It

```
class A
{
A()                  // Function 1 A x;
A(int a)             // Function 2 A x(4);
A(int a,int b)       // Function 3 A x(5,3);
A(int a,int b,int c) // Function 4 A x(3,4,5);
A(int a,char b)      // Function 5 A x(4,'A');
A(char a,int b)      // Function 6 A x('A',5);
A(char a,char b)     // Function 7 A x('A','B');
A(char a[ ],char b)  // Function 8 A x("ABC",4);
A(char a,char b[])   // Function 9 A x(5,"ABC");
public:

A(char a[ ],char b[])// Function 10 A x("ABC","XYZ");
A(int  a,char b[],int c) // Function 11 A x(5,"ABC",4);
A(int a,char b,char c[]) // Function 12 A x(4,5,"ABC");
A(A &X)                  // Function 13 A x(4); A y(x);
void Add(int a , int b) // Function 14
 void Mul(int a,int b,int c)// Function 15;
A x;
x.Add(4,5);              // Function 14
x.Add(4,5,6);            // Function 15;
```

Play with C++

Solve Question

Given the following C++ code, answer the questions (i) & (ii).

```
class TestMeOut
{public :
~TestMeOut() // Function 1
{ cout << "Leaving the examination hall " << endl; }
TestMeOut() // Function 2
{ cout << "Appearing for examination " << endl; }
void MyWork() // Function 3
{
cout << "Attempting Questions " << endl;
 }
};
```

(*i*) In Object Oriented Programming, what is Function 1 referred as and when does it get invoked/ called ? : Destructor

(*ii*) In Object Oriented Programming, what is Function 2 referred as and when does it get invoked/ called ?: Constructor

Answer the questions after going through the following class code

Q1. class Exam

```
{char Subject[20] ;
int Marks ;
public :
Exam()                          // Function 1
{
strcpy(Subject, “Computer” ) ;
Marks = 0 ;
}
Exam(char P[ ])                 // Function 2
{
strcpy(Subject, P) ;
Marks=0 ;
}
Exam(int M)             // Function 3
{
strcpy(Subject, “Computer”) ;
Marks = M ;
}
Exam(char P[ ], int M)   // Function 4
{
strcpy(Subject, P) ;
Marks = M ;}};
```

a) Which feature of the Object Oriented Programming is demonstrated using Function 1, Function2, Function 3 and Function 4 in the above class Exam?

b)Write statements in C++ that would execute Function 3 and Function 4 of class Exam.

Q2. Class TEST

```
{
int time;
 public:
TEST( )                         //Function 1
{
time=0;
cout<< ”hai”;
}
~ TEST( )               //Function 2
{
cout<< ”hello”;
}
void exam( )                    //Function 3
{
cout<<”god bless u”;
}
TEST(int Duration)              //Function 4
{
time=Duration;
cout<<”Exam starts”;
}
TEST(TEST &T)                   //Function 5
{
time = T.Duration;
cout<<”Exam Finished”
}
};
```

(a) In Object Oriented Programming, what is Function 1 referred as and when does it get invoked/called?

(b) In Object Oriented Programming, what is Function 2 referred as and when does it get invoked / called?

(c) Which category of constructors does Function 5 belong to and what is the purpose of using it?

(d) Write statements that would call the member Function 1 and 4.

Q3.class

```
class Exam
{
 int year;
public:
Exam(int y)       //Constructor 1
{
year=y;
}
Exam(Exam & t);          //Constructor 2
};
```

i. Create an object, such that it invokes Constructor 1.

ii. Write complete definition for Constructor 2.

Q.4 class student

```
{
 int rno;
 char name[20];
 float per;
student()                       // Function 1
{
rno=1;
 strcpy(name,"Raman");
 per=85.5;
 cout<<”Student is initialized “<<endl  }
public:
 void showdata()                // Function 2
```

Play With C++

```
 {
 cout<<name<<”scored“<<per<<” %marks”<<endl; }
 ~student ()                    // Function 3
 {
cout<<”object is destroyed “<<endl;    }
};
void main()
{  student S;              //Statement 1
 S.showdata();             //Statement 2
}
```

i. Will Statement -1 initialize all the data members for object S with the values given in the Function 1 ? (Yes or No). Justify your answer suggesting the correction(s) to be made in the above code.

ii. What shall be the possible output after the execution of program? (Assuming , if required the suggested correction(s) are made in the program)

Q5. class mammal{

```
public:
char category[20];
mammal( char xname[]) // function1
{
strcpy(category, xname)
}
mammal(mammal &t); //function2
};
```

(i) Create an object, such that it invokesfunction1.

(ii) Write complete definition for function2.

Q6.class TestMeOut

```
{
public :
~TestMeOut() // Function 1
{ cout << "Leaving the examination hall " << endl; }
TestMeOut() // Function 2
{ cout << "Appearing for examination " << endl; }
void MyWork() // Function 3
{ cout << "Attempting Questions " << endl; }
};
```

(*i*) In Object Oriented Programming, what is Function 1 referred as and when does it get invoked/ called ?

(*ii*) In Object Oriented Programming, what is Function 2 referred as and when does it get invoked/ called ?

Q7.class schoolbag

```
{int pockets;
public:
schoolbag() //Function 1
{ pockets=30;
cout<<"The bag has pockets"<<endl;
}
void company() //Function 2
{
cout<<"The company of the Bag is ABC"<<endl;}
schoolbag(int D) //Function 3
{
pockets=D;
cout<<"Now the Bag has pockets"<<pockets<<endl;}
~schoolbag() //Function 4
{
cout<<"Thanks"<<end1;}};
```

(i) In Object Oriented Programming, what is Function 4 referred as and when does it get invoked/called ?

(ii) In Object Oriented Programming, which concept is illustrated by Function 1 and Function 3 together ?

Q3 . Answer the questions after going through the following class.

```
class Exam
{
 char Subject[20] ;
int Marks ;
public :
Exam()                      // Function 1
{
strcpy(Subject, “Computer” ) ;
Marks = 0 ;
}
Exam(char P[ ])                    // Function 2
{
strcpy(Subject, P) ;
Marks=0 ;
}
Exam(int M)                 // Function 3
{
strcpy(Subject, “Computer”) ;
Marks = M ;
}
Exam(char P[ ], int M)   // Function 4
{
strcpy(Subject, P) ;
Marks = M ;}};
```

a) Which feature of the Object Oriented Programming is demonstrated using Function 1, Function2, Function 3 and Function 4 in the above class Exam?

b) Write statements in C++ that would execute Function 3 and Function 4 of class Exam.

Q4. Given the following C++ code, answer the questions

```
class TEST
{           int time;
public:
TEST( )            //Function 1
{time=0;
cout<< ”hai”;}
~ TEST( ) //Function 2
{           cout<< ”hello”;}
void exam( )                 //Function 3
{
```

Play with C++

```
cout<<"god bless u";}
TEST(int Duration)      //Function 4
{time=Duration;
cout<<"Exam starts";}
TEST(TEST &T)           //Function 5
{           time = T.Duration;
cout<<"Exam Finished"}};
```

(a) In Object Oriented Programming, what is Function 1 referred as and when does it get invoked/called?

(b) In Object Oriented Programming, what is Function 2 referred as and when does it get invoked / called?

(c) Which category of constructors does Function 5 belong to and what is the purpose of using it?

(d) Write statements that would call the member Function 1 and 4.

Revision test Constructor Destructor

Q1.Define With Example: (any 2)

- Copy Constructor
- Default Constructor and Parameterize Constructor
- Constructor and Destructor

Q2. Can we declare a constructor in private section ? Give reasons.

Q3. Answer the questions after going through the following class code

```
class mammal
{
public:
mammal();                  // function1
mammal(int a);                    // function2
mammal( char xname[]) // function3
mammal(mammal &t); //function4
void x();                         // function5
void x(int a);             // function6
void x(int a,int b);       // function7
~mammal();                 //function 8};
```

(i) Create an object, such that it invokes **function1 to Function 7**.

(ii) Write complete definition for function4 and function 8

(iii) which technology of OOPS in Function 1 to 4 and **5 to 7**

(iv) When default constructor and destructor call

Q4. (ii) Answer the following :

```
CLASS ABC
{ int a,b;     char ch;
  PUBLIC :
 ABC()                      // fun1
{ cout<<" constructor is working";}
ABC( int i)                 // fun2
{ a= i;}
 ABC(int  j, char g)        // fun3
{ int b=j;
  ch=g; }
ABC(  ABC o)                //fun4
 { a=o.a;     b=o.b;     ch=o.ch; }
~ ABC()                     //fun5
  { cout<<"destructing";} }
```

(i) Write the particular names of function 1,2,3 and when they are invoked ?

(ii) Write the statements to call function 1,2,3

(iii) What is fun4 & write the statement to call fun4

(iv) What is func 5 when does it invoked what is the concept behind all these function4

Play with C++

Q5) Give the output of the following program :

```
# include <iostream.h>
# include <string.h>
class per
{char name [20];
float age;
public:
per (char*s, float a)
{Strcpy(name, s);
age = a;}
per&GR (per &x)
{if (x.age>= age)
return x;
else
return *this;}
void display()
{cout<<”Name :” << name << '\n';
cout<< “Age :” << age << '\n';}};
void main ()
{per P1 ("RAMU", 27.5), P2 ("RAJU", 53), P3("KALU", 40);
per P ('\0', 0);
P = Pl.GR (P3);
P.display();
P = P2.GR (P3);
P.display();}
```

Q6) Answer the questions (i) and (ii) after going through the following class

```
classMaths
{
char Chapter [20];
int Marks;
public:
Maths ( )
{
strcpy (Chapter, “Geometry”);
Marks = 10;
cout<<“Chapter Initialised”;
{
~Math ( ) //Member Function 2
}
cout<<”Chapter Over”;
}};
```

(i) Name the specific features of class shown by Member Function 1 andMember Function 2 in the above example.

(ii) How would Member Function 1 and Member Function 2 get executed?

MODEL 2:

Answer the questions (i) and (ii) after going through the following class 2Marks

1.Answer the questions(i) and (ii) after going through the following class: (2017 MP)

```
class planet
{
char name[20];char distance[20];
public:
planet() //Function 1
{
strcpy(name, "Venus");
strcpy(distance,"38 million km");
}
void display(char na[],char d[]) //Function 2
{
cout<<na<<"has"<<d<<"distancefromEarth"<<endl;
}
planet(char na[], char d[]) //Function 3
{
strcpy(name,na);
strcpy(distance,d);
}
~planet() //Function 4
{
cout<<"Planetarium time over!!!"<<endl;
}
};
```

I. What is Function 1 referred as? When will it be executed?

II. Write suitable C++ statement to invoke Function 2.

Ans I. Constructor. It will be executed at the time of object creation.

II. planet p;
p.display(“Pluto”,”7.5 Billion Km”);

2) Observe the following C++ code and answer the questions (i) and (ii). Assume all necessary files are included: (2016)

```
class BOOK
{
long Code ;
char Title[20];
float Price;
public:
BOOK() //Member Function 1
{
cout<<”Bought”<<endl;
Code=10;strcpy(Title,”NoTitle”);Price=100;
}
BOOK(int C,char T[],float P) //Member Function 2
{
```

```
Code=C;
strcpy(Title,T);
Price=P;
}
void Update(float P) //Member Function 3
{
Price+=P;
}
void Display() //Member Function 4
{
cout<<Code<<”:”<<Title<<”:”<<Price<<endl;
}
~BOOK() //Member Function 5
{
cout<<”Book Discarded!”<<end1;
}
};
void main() //Line 1
{ //Line 2
BOOK B,C(101,”Truth”,350}; //Line 3
for (int I=0;I<4;I++) //Line 4
{ //Line 5
B.Update(50);C.Update(20); //Line 6
B.Display();C.Display(); //Line 7
} //Line 8
} //Line 9
```

(i) Which specific concept of object oriented programming out of the following is illustrated by Member Function 1 and Member Function 2 combined together?

- ● Data Encapsulation
- ● Polymorphism
- ● Inheritance
- ● Data Hiding

Ans Polymorphism

(ii) How many times the message ”Book Discarded!” will be displayed after executing the above C++ code? Out of Line 1 to Line 9, which line is responsible to display the message ”Book Discarded!”

Ans 2 times Line 9

3)Observe the following C++ code and answer the questions (i) and (ii) : (2015)

```
class Passenger
{
long PNR;
char Name [20] ;
public:
Passenger( ) //Function 1
{ cout<<"Ready"<<endl; }
void Book(long P,char N[]) //Function 2
{ PNR = P; strcpy(Name, N); }
void Print() //Function 3
{ cout«PNR << Name <<endl; }
~Passenger() //Function 4
{ cout<<"Booking cancelled!"<<endl; }
};
```

(i) Fill in the blank statements in Line 1 and Line 2 to execute Function 2 and Function 3 respectively in the following code:

```
v oid main()
{
Passenger P;
__________ //Line 1
__________ //Line 2
}//Ends here
```

Ans P.Book(1234567,”Ravi”); //Line 1
P.Print(); //Line 2

(ii) Which function will be executed at }//Ends here? What is this function referred as ?

Ans Function 4 OR ~Passenger(). It is a Destructor function.

4) Answer the questions (i) and (ii) ater going through the following class: (2014)

```
class Hospital
{
int Pno,Dno;
public:
Hospital(int PN); //Function 1
Hospital( ); //Function 2
Hospital (Hospital &H); //Function 3
void In( ); //Function 4
void Disp ( ); //Function 5
};
void main( )
{ Hospital H(20); //Statement 1
}
```

(i) Which of the function out of function 1,2,3,4 or 5 will get executed when the statement 1 is executed in the above code?

A) Function 1 will be executed when the statement 1 is executed.

(ii) Write a statement to declare a new object G with reference to already existing object H using Function 3.

A) Hospital G(H);

5) Answer the questions (i) and (ii) after going through the following class: (2013)

```
class Race
{
int CarNo,Track;
public:
Race( );//Function 1
Race(int CN);//Function 2
Race(Race &R);//Function 3
void Register( );//Function 4
void Drive( );//Function 5
};
void main( )
{
Race R;
----
----
}
```

(i) Out of the following, which of the option is correct for calling Function 2?

(a) Option 1 – Race T(30);

(b) Option 2 – Race U(R);

Ans) (a) Option 1 – Race T(30);

(ii) Name the feature of Object Oriented Programming which is
illustrated by Function 1, Function 2 and Function 3 combined
together.

Anser) Constructor Overloading.

6) Answer the questions (i) and (ii) after going through the following class (2012)

```
class Travel
{
int PlaceCode; char Place[20] ; float Charges;
public:
Travel () //Function 1
{
PlaceCode=l;strcpy (Place, "DELHJ:") ; Charges = 1000;
}
void TravelPlan (float C) //Function 2
{
cout<<PlaceCode<<":"<<Place«":"<<Charges<<endl;
}
~Travel ( ) //Function 3
{
Cout<<"Travel Plan Cancelled"<<endl;
}
Travel (int PC, char P[], float C) //Function 4
{
PlaceCode=PC;strcpy(Place,P); Charges=C;
}
} ;
```

(i) In Object Oriented Programming, what are Function 1 and Function 4 combined together referred as?

Ans (i) Polymorphism OR Constructor Overloading
OR Overloaded Constructor OR Function Overloading
OR Overloaded Functions OR Default Constructor and
Parameterized Constructor

(ii) In Object Oriented Programming, which concept is illustrated by Function 3? When is this function called invoked?

Ans. (ii) Destructor. It is called / Invoked when an object of the class goes out of scope. Note:*(Full 1 Mark to be given if only the correct invocation is written)*

7) Write the output of the following C++ code. Also, write the name of feature of Object Oriented Programming used in the following program jointly illustrated by the function [I] to [IV]. 2011

```
#include<iostream.h>
void Line ( ) //Function [I]
{
for (int L=1;L<=80;L++) cout<<"-";
cout<<end1;
}
void Line (int N) //Function[II]
{
for (int L=l;L<=N;L++) Cout<<"*";
cout<<endl;
}
void Line (char C, int N) //Function [III]
{
for (int L=l;L<=N;L++) cout<<C;
cout<<end1;
}
void Line (int M, int, N) //Function [IV]
{
for (int L=1;L<=N;L++) cout<<M*L;
cout<<end1;
}
void main ( )
{
int A=9, B=4, C=3;
char K= '#' ;
Line (K,B);
Line (A,C);
}
```

Ans

```
####
91827
```

Polymorphism OR Function Overloading

(8) Answer the questions (i) and (ii) after going through the following class: (2010D)

```
class TEST
{ int Regno, Max, Min, Score;
public:
TEST() //Function 1
{
Regno= 101;
Max=100;
Min=40;
Score=75;
}
TEST(int Pregno,int Pscore) //Function 2
{
Regno=Pregno;
Max=100;
Min=40;
```

Play with C++

```
Score=Pscore;
}
~TEST() //Function 3
{
cout<<“TEST Over”<<endl;
}
void Display() //Function 4
{
cout<<Regno<<“:”<<Max<<“:”<<Min<<endl;
cout<<“[Score]”<<Score<<endl;
}
};
```

(i) As per Object Oriented Programming, which. concept is illustrated by **Function 1 and Function 2** together?

Ans. Polymorphism (OR) Function Overloading
(OR) Constructor Overloading

(ii) What is **Function 3** specifically referred as ? When do you
think, **Function 3** will be invoked/called?

Ans.
Destructor, invoked or called when scope of an Object

9) Answer the questions (i) and (ii) after going through the following class: (2010OD)

```
class Exam
{
int Rno,MaxMarks,MinMarks,Marks;
public:
Exam ( ) //Module 1
{
Rno=101;
MaxMarks=l00;
MinMarks=40;
Marks=75;
}
Exam (int Prno, int Pmarks) //Module 2
{ Rno=Prno;
MaxMarks=l00;
MinMarks=40;
Marks=Pmarks;
}
~Exam () //Module 3
{
cout<<“Exam Over”<<endl;
}
void Show () //Module 4
{
cout<<Rno<<“:”<<MaxMarks<<“:”<<MinMarks<<endl;
cout<<“[Marks Got]”<<Marks<<endl;
}
};
```

(i) As per Object Oriented Programming, which concept is illustrated by**Module 1** and **Module 2** together?

Ans. Polymorphism (OR) Constructor Overloading (OR) Function Overloading

(ii) What is **Module 3** referred as ? When do you think, **Module 3** will be invoked/called?

Ans. Destructor. It is invoked as soon as the scope of the object gets over.

(10) Answer the questions (i) and (ii) after going through the following class: (2009D)

```
class WORK 2
{
int WorkId;char WorkType ;
public:
-WORK ( ) //Function 1
{
cout<<”Un-Allocated”<<endl ;
}
void status ( ) //Function 2
{
XII Computer Chap 4 to6 5 0
cout<<WorkId<<”: “<<WorkType<<endl ;
}
WORK ( ) //Function 3
{
WorkId = 10;
WorkType=’T’ ;
}
WORK(WORK &W) //Function 4
{
WorkId=W. WorkId+12;
WorkType=W. WorkType+l
}
} ;
```

(i) Which member function out of Function 1, Function 2, Function 3 and Function 4 shown in the above definition of class WORK is called automatically, when the scope of an object gets over? Is it known as Constructor OR Destructor OR Overloaded Function OR Copy Constructor?

Ans Function 1 Destructor.

(ii) WORK W; // Statement 1
WORK Y(W); // Statement 2
Which member function out of Function 1, Function 2, Function 3 and Function 4 shown in the above definition of class WORK will be called on execution of statement written as statement 2 ? What is this function specifically known as out of Destructor or Copy Constructor or Default Constructor?

Ans Function 4 Copy Constructor.

(11) Answer the questions (i) and (ii) after going through the following class: (2009 OD)

```
class Job
{int JobId;
char JobType;
public:
~Job ( ) //Function 1
```

Play with C++

Play with C++

```
{
cout<< "Resigned" <<end1;
}
Job ( ) //Function 2
{ JobId=10 ;
JobType ='T" ;
}
void TellMe( )//Function 3
{
cout<<JobId<< ": " <<JobType<<end1;
}
Job (Job &J) //Function 4
{
JobId=J.JobId+10;
JobType=J.JobType+l;
}
};
```

(i) Which member function out of Function 1, Function 2, Function 3 and Function 4 shown in the above definition of class Job is called automatically, when the scope of an object gets over? Is it known as Constructor OR Destructor OR Overloaded Function OR Copy Constructor?

Ans Function 1. Destructor.

(ii) Job P ; //Line 1 Job Q(P) ; //Line 2

Which member function out of Function 1, Function 2, Function 3 and Function 4 shown in the above definition of class Job will be called on execution of statement written as Line 2 ? What is this function specifically known as out of Destructor or Copy Constructor or Default Constructor?

Ans Function 4. Copy Constructor.

12) Answer the questions (i) and (ii) after going through the following program: (2008D)

```
#include <iostream.h>
#include<string.h>
class bazaar
{ char Type[20] ;
char product [20];
int qty ;
float price ;
bazaar() //function 1
{ strcpy (type , "Electronic") ;
strcpy (product , "calculator");
qty=10;
price=225;
}
public :
void Disp() //function 2
{ cout<< type <<"-"<<product<<":"
<<qty<< "@" << price << endl ;
}
};
void main ()
{ Bazaar B ; //statement 1
B. disp() ; //statement 2
}
```

(i) Will statement 1 initialize all the data members for object B with the values given in the function 1 ? (YES OR NO). Justify your answer suggesting the correction(s) to be made in the above code.

Ans: No. The reason is the constructor should be defined under the public visibility label.

(ii) What shall be the possible output when the program gets executed ? (Assuming, if required _ the suggested correction(s) are made in the program).

Ans: Possible Output:

Electronic–Calculator:10@225

13) Answer the questions (i) and (ii) after going through the following program: (2008OD)

```
#include<iostream.h>
#include<string.h>
class Retail
{ char category[20];
char item[20];
int qty;
float price;
retail () //function 1
{ strcpy (category, "cerial");
strcpy (Item, "Rice");
qty =100 ;
price =25 ;
}
public;
void show() //function 2
{ cout << category <<"-"<< Item << "
:"<<Qty<<"@"<< price<<endl;
}
};
void main()
{ Retail R; //statement 1
R. show (); //statement 2
}
```

(i) will statement 1 initialize all the data members for objects R with the given in the function 1 ? (YES OR NO). Justify your Answer suggesting the corrections(s) to be made in the above code.

Ans: No. The reason is the constructor should be defined underthe public visibility label.

(ii) What shall be the possible out put when the program gets executed ? (Assuming, if required the suggested correction(s) are made in the program)

Ans: Possible Output:

cerial–Rice:100@25

14) Answer the question (i)and (ii)after going through the following class: (2007D)

```
class Maths
```

```
{ char Chapter[20]
int Marks;
public:
Maths() //Member Function 1
{ strcpy (Chapter, “Geometry”);
Marks=10;
cout <<”Chapter Initialised “;
}
-Maths() //Member Functions 2
{ cout<<”Chapter Over”;
}
};
```

(i) Name the specific features of class shown by member Function 1 and Member Function 2 in the above example.

Ans: Member function 1 is a (non-parameterized or default) constructor (, which will be executed automatically at the time of creation of an object of class Maths). Member function 2 is a destructor (,which will be executed automatically at the time of destruction of an object of class Maths).

(ii) How would Member Function 1 and Member Function 2 get executed ?

Ans: They will be executed automatically. Member function 1 will be executed at the time of creation of an object of class Maths. Member function 2 will be executed at the time of destruction of an object of class Maths.

15) Answer the questions (i) and (ii) after going through the following class : (2007OD)

```
class Science
{ char Topic[20] ;
int Weightage ;
public :
Science () //Function 1
{ strcpy (Topic, “Optics”) ;
Weightage =30
cout<<”Topic Activated”;
}
~Science() //Function 2
{ cout<<”Topic Deactivated”; }
};
```

(i)Name the specific features of class shown by Function 1 and Function 2 in the above example.

Ans: Member function 1 is a (non-parameterized or default) constructor (, which will be executed automatically at the time of creation of an object of class Science). Member function 2 is a destructor (,which will be executed automatically at the time of destruction of an object of class Science).

(ii) How would Function 1 and Function 2 get executed ?

Ans: They will be executed automatically.

Member function 1 will be executed at the time of creation of an object of class Science. Member function 2 will be executed at the time of destruction of an object of class Science.

16) Answer the following questions (i) and (ii) after going through the following class. (2006D)

```
class Interview
{ int Month;
public:
interview(int y) {Month=y;}//constructor 1
interview(Interview&t); //constructor 2
};
```

(i) create an object, such that it invokes Constructor 1.

Ans: Interview A(10); //invoking constructor 1 by passing a number.

(ii) write complete definition for Constructer 2.

Ans: Interview(Interview &t)

```
//This is a copy constructor.
{ Month=t.Month;}
```

17b) Answer the following questions (i) and (ii) after going through the following class. (2006 OD)

```
class Exam
{ int Year;
public:
Exam(int y) //Constructor 1
{ Year=y;
}
Exam(Exam &t);
//Constructor 2
};
```

(i) Create an object, such that it invokes Constructor 1

Ans: Exam E((2008);

(ii) Write complete definition for constructor 2.

Ans: Exam(Exam &t)

```
//Copy Constructor.
{ Year=t.Year;
}
```

18) Answer the following questions (i) and (ii) after going through the following class. (2005D)

```
class Test
{ char Paper[20];
int Marks
public:
Test() //Function 1
{ strcpy(Paper,”Computer”);
Marks=0;
}
Test(char P[])
{ strcpy(Paper,P);
Marks=0;
}
Test(int M)
{ strcpy(Paper,”Computer”);
Marks=M;
}
```

Play with C++

```
Test(char P[],int M)
{ strcpy(Paper,P);
Marks=M;
}
};
```

(i)Which feature Object Oriented programming is demonstrated using Function 1, Function 2, Function 3 and Function 4 in the above class text?

Ans: Function overloading (here it is constructor overloading).

(ii)Write statements in C++ that would execute Function 2 and Function 4 of class Text.

Ans: (let char name[20];

```
int X=60;
strcpy(name,"COMPUTERSCIENCE");
```

are declared in the program)

(i) Test A(name); //Will execute Funciton 2

(ii) Test B(name,X); //Will execute Function 4

19) Answer the following questions (i)and (ii) after going through the following class. (2005 OD)

```
class Exam
{ int Marks;
char Subject[20];
public:
Exam() //Function 1
{ strcpy(Subject,"Computer");
Marks=0;
}
Exam(char S[]) //Function 2
{ strcpy(Subject,S);
Marks=0; }
Exam(int M) //Function 3
{ strcpy(Subject,"Computer");
Marks=M;
}
Exam(char S[],int M) //Function4
{ Strcpy(Subject,P);
Marks=M;
}
};
```

(i)Write statements in C++ that would execute Function 3 and Function 4 of class Exam.

(let char name[20];

int X=60;

strcpy(name,"COMPUTERSCIENCE"); are declared in the program)

(i) Exam A(X); //Will execute Funciton 3

(ii) Exam B(name,X); //Will execute Function 4

(ii)Which feature Object Oriented Programming is demonstrated using Function 1, Function 2, Function 3 and Function 4 in the above class text?

Ans: Function overloading (here it is constructor overloading).

20)Given the following C++ code, answer the questions (i)and(ii) (2004 D)

```
class TestMeOut
{ public:
~TestMeOut( ) //Function 1
{
cout<<"Leaving the examination hall"<<endl;
}
TestMeOut( ) //Function 2
{
cout<<"Appearing for examination"<<endl;
}
void MyWork( )
{
cout<<"Attempting Questions"<<endl;
}
};
```

(i) In Object Oriented programming, what is Function 1 referred as and when does it get invoked/called?

Ans: Function 1 is called as Destructor, It will automatically executed at the time of destruction of the object of class TestMeOut.

(ii) In Object Oriented Programming, what is Function 2 referred as and when does it get invoked/called?

Ans: Function 2 is called as constructor (Non-parameterized or default constructor) , it will automatically executed at the time of creation of the object of class TestMeOut.

21)Answer the questions (i) and (ii) after going through the following class: (2008-09MP1)

```
class Seminar
{
int Time;
public:
Seminar() //Function 1
{
Time=30;cout<<"Seminar starts now"<<end1;
}
void Lecture() //Function 2
{
cout<<"Lectures in the seminar on"<<end1;
}
Seminar(int Duration) //Function 3
{
Time=Duration;cout<<"Seminar starts
now"<<end1;
}
~Seminar() //Function 4
{
cout<<"Vote of thanks"<<end1;}};
```

i)In Object Oriented Programming, what is Function 4 referred as and when does it get invoked/called?

Answer: Destructor, it is invoked as soon as the scope of the object gets over.

ii) In Object Oriented Programming, which concept is illustrated by Function 1 and Function 3 together? Write an example illustrating the calls for these functions.

Answer:

Constructor Overloading (Polymorphism)

Seminar S1,S2(90);

22)Answer the questions (i) and (ii) after going through the following program (2008-09 MP2)

```
class Match
{
int Time;
public:
Match() //Function 1
{
Time=0;
cout<<”Match commences”<<end1;}
void Details() //Function 2
{
cout<<”Inter Section Basketball Match”<<end1;}
Match(int Duration) //Function 3
{
Time=Duration;
cout<<”Another Match begins now”<<end1;}
Match(Match &M) //Function 4
{
Time=M.Duration;
cout<<”Like Previous Match ”<<end1;}};
```

i)Which category of constructor - Function 4 belongs to and what is the purpose of using it?

Answer:

Copy Constructor, it is invoked when an object is created and initialised with values of an already existing object.

*ii)*Write statements that would call the member Functions 1 and 3

Answer:

Match M1; //for Function 1

Match M2(90); //for Function 3

23) Answer the questions (i) and (ii) after going through the following class: (2009-10 MP1)

```
class Seminar
{ int Time;
XII Computer Chap 4 to6 5 3
public:
Seminar() //Function 1
{
Time=30;
cout<<"Seminar starts now"<<end1;
}
void Lecture() //Function 2
{
cout<<"Lectures in the seminar on"<<end1;
}
Seminar(int Duration) //Function 3
{ Time=Duration;
cout<<"Seminar starts now"<<end1;
}
~Seminar( )//Function 4
{ cout<<"Vote of thanks"<<end1;}};
```

i) In Object Oriented Programming, what is Function 4 referred as and when does it get invoked/called?

A) Destructor, it is invoked as soon as the scope of the object gets over.

ii) In Object Oriented Programming, which concept is illustrated by Function 1 and Function 3 together? Write an example illustrating the calls for these functions.

A) Constructor Overloading (or Function Overloading or Polymorphism)

Seminar S1; //Function 1

Seminar S2(90); //Function 3

24) Answer the questions (i) and (ii) after going through the following program: (2009-10 MP2)

```
class Match
{
int Time;
public:
Match() //Function 1
{
Time=0;
cout<<"Match commences"<<end1;}
void Details() //Function 2
{
cout<<"Inter Section Basketball
Match"<<end1;}
Match(int Duration) //Function 3
{
Time=Duration;
cout<<"Another Match begins
now"<<end1;}
Match(Match &M) //Function 4
{
Time=M.Duration;
cout<<"Like Previous Match "<<end1;}
};
```

i) Which category of constructor - Function 4 belongs to and what is the purpose of using it?

A) Copy constructor, It will help to copy the data from one objectto another

ii) Write statements that would call the member Functions 1 and 3 A) Match M; //Function 1Match N(10);
//Function 3

Play With C++