

Assignment - 13

PAGE No.	/ /
DATE	/ /

Q1) what is meant by reference in C++?
Explain with an example

- reference is newly added datatype in C++
- reference is considered as derived datatype in C++
- when we create a reference to an existing variable it is just considered as another name to that variable
- when we create reference there is no memory allocation for it.
- when we create reference variable its entry gets added inside symbol table.

Example:-

```
#include <iostream>
using namespace std;
int main()
{
    int no = 11;
    int noref = no;
    cout << "no::" << no << "\n";
    cout << "noref::" << noref << "\n";
    return 0;
}
```

(Q 2) what is mean by constructor & destructor when are they get called?

Constructor:-

- Constructor is a special function which gets automatically call when create the object of class.
- Compiler will call constructor before allocating the memory for variable.
- following are types of constructor

- 1) Default constructor
- 2) Parametrized constructor
- 3) copy constructor.

Destructor:-

- Destructor is also special function.
- Destructor is a function which gets call before deallocating memory for an object.

Q8) write down the rules which must be followed while writing constructor & destructor

Rules for constructor

- 1) name of constructor will be same as class name
- 2) constructor will be inside public Access specifier
- 3) there should not any returntype or datatype for constructor.
- 4) No need of explicit call to the constructor

Rules for destructor

- 1) name of destructor will be same as class name ; but with a tilde (~) operator in front.
- 2) destructor will be inside public Access specifier
- 3) there should not any returntype or datatype for destructor.
- 4) No need to explicit call to the destructor

(Q) what are different types of constructor? explain with sample program

following are types of constructor

1) Default constructor -

If we create an object without passing parameter then the default constructor will call

e.g:-

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

```
class Demo
```

```
{
```

```
int i;
```

```
public:
```

```
Test()
```

```
{
```

```
i = 10;
```

```
}
```

```
};
```

2) Parametrized constructor

- If we create an object by passing parameter the that constructor will call parametrized constructor.

e.g:-

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

```
class Demo
```

```
{
```

```
public :
```

```
int i;
```

```
Demo( int j )
```

```
{
```

```
j = j;
```

```
}
```

```
}
```

```
};
```

```
};
```

3) Copy Constructor

If we create an object by the another object then the copy constructor gets call.

e.g:-

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

```
class Demo
```

```
{
```

`cout << "value of i = " << obj.i << endl`

`public:`

`int i;`
`int j;`

`Demo (Demo& ref)`

`{`

`i = ref.i;`

`j = ref.j;`

`cout << "inside copy constructor"`

`}`

`};`

Q5) what is mean by this pointer? What is the prototype of this pointer

- In case of C++ & Java to call any non-static function we need object of class

- By using object . & operator we can call the any obj.

- when we use that caller object internally compiler will sends the Address of that object as first hidden explicit parameter.

- that Address gets stored inside first explicit of that Argument & that Argument is called this pointer.
prototype:

This → member name;

this → i = 10;

Q6) what is inheritance? explain type
of inheritance with program

→ inheritance consider as one of the
object oriented paradigm in C++.

- In the simple term inheritance is
defines: reusability.

- By using concept of inheritance class
Acquire properties & behaviors of class.

i) single level inheritance.

- In single level inheritance parent &
child class are present.

e.g.:

```
#include <iostream>
using namespace std;
class Demo { Parent class
{
public x, y;
```

void fun ()

```
{ cout << "inside fun of demo class" <<
endl;
```

}

};

```
class Hello :: public Demo
{
public:
    int a,b;
};

void fun()
{
    cout<<"inside Hello fun:"<<endl;
}

int main()
{
    Hello obj();
    obj.fun();
    obj.fun();
}
```

3) multilevel inheritance

- It is considered as extension of single level inheritance.
 - In case of multilevel inheritance there should be atleast 3 class

e.g.:-

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

```
class Demo
```

```
{ public:
```

```
    int x, y;
```

```
    void func()
```

```
{
```

```
    cout << "inside fun of Demo:" << endl;
```

```
}
```

```
}
```

```
// child class
```

```
class Hello : public Demo
```

```
{
```

```
public:
```

```
    int a, b, c;
```

```
    void gun()
```

```
{
```

```
    cout << "inside gun of hello" << endl;
```

```
}
```

```
,
```

```
class JSM : public Hello
```

```
{
```

```
public:
```

```
    int p;
```

```
    void sun()
```

```
{
```

```
    cout << "inside sun of JSM" << endl;
```

```
}
```

```
int main()
{
    JSM obj();
    cout << JSMobj.x << endl;
    cout << JSMobj.y << endl;
    cout << JSMobj.a << endl;
    cout << JSMobj.b << endl;
    cout << JSMobj.c << endl;
    cout << JSMobj.p << endl;

    JSM obj.fun();
    JSM obj.gun();
    JSM obj.sun();

    return 0;
}
```

3) multiple inheritance

In this type of inheritance one class can inherit more than one class at a time

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

```
class Demo
{
public:
    int int a;
    void func();
```

class Hello

{
void sun () { }
}

class JSM: public Demo, public Hello

{
void mun ()
{
}

int main ()

{
JSM obj;
obj.fun();
obj.sun();
obj.mun();

{
return 0;
}

Q7) what is mean by polymorphism
Explain its types with program

→ the term polymorphism is defined as single name of multiple behaviour.

- the concept of polymorphism is present every object oriented program with same technical change.

- there are 2 types of polymorphism

1) compile time polymorphism

- The concept of compile time polymorphism is considered early binding because the decision taken by compiler is in early phase.

- to Achieve compile time polymorphism in C++ we need the term called as overloading.

e.g:-

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

```
class Arithmetic
```

```
public:
```

```
    int Addition (int n1, int n2)
```

```
    {
```

2) `int Addition (int no1, int no2, int no3)`
 ³
 ₃

`int Addition (float no1, float no2)`
 ³
 ₃

3) Runtime polymorphism

- This type of polymorphism is
Achieved by ~~over~~ function overriding

- it is also called as late binding.

- function calling decision is taken
at runtime.

e.g:-

```
#include <iostream>
using namespace std;
class parent
{
public:
    void fun ()
    {
        cout << "inside fun" << endl;
    }
};
```

```
class child : Public Parent  
{  
public :  
    void Run ()  
    {  
        cout << "inside Run." << endl;  
    }  
};  
  
int main()  
{  
    child obj();  
    obj.fun();  
    obj.Run();  
    return 0;  
}
```

- Q8) what is the use of & operator in case of copy constructor
- In C++ ampersand (&) operator is used in a copy constructor to pass an object by reference.
 - A copy constructor creates a new separate copy of an existing object
- Q9) why the name of constructor and destructor is same as class name?
- In the C++ the constructor & destructor have the same name as the class because the constructor creates the object and the destructor destroys it.
 - A constructor & destructor is a special member function that initialize the objects of its class. It is special because its name is same as the class name.

Ques 10) what is mean by overloading? Explain function overloading

→ overloading concept comes under the compile time polymorphism.

- there are 2 type of overloading

1) function overloading

2) operator overloading

- To Achieve the compile time polymorphism in C++ & Java we use the overloading concept.

In case of function overloading we can define multiple function in a class with same name & different prototype (parameters)

- The compiler chooses which function to use based on the number and type of arguments passed.

- function overloading takes place within a one class