

Name: Ajinkya
Rathod

Subject: OOP

Assignment: 2

Assignment - 2

1. What is constructor?

Ans Constructor is a method in a class which is invoked automatically when we declare a object.

Explicit Constructor

When a constructor is not stated, compiler automatically calls constructor.

But sometimes we need to give initial values we need explicit constructor for that.

Parameterized Constructor

The constructor having parameters is called Parameterized constructor.

or Explain copy constructor?

Ans Assignment operator works on primitive datatypes. what if we want it in user-defined datatype i.e. class.

So, we used copy constructor.

#include <iostream>

using namespace std;

class point

int x, y;

public:

point (int a, int b) :
x(a), y(b)

x(a);

x(b);

3

void display() {

cout << a << b;

4;

```
int main() {  
    point p1(1, 1);  
    p1.display();  
    cout << p1(100);  
    cout << EA;
```

Q3 Explain M16

1. Member wise initialisation
of a class using constructor
function,
a. It starts with ; od has all
3 fields separated by comma
2. The initialization has a normal
syntax of initialization
; a (-value) value).

eg $\text{ff} \in \text{de} < \text{posten}$
using memory slot;

thus Time t

parts :-

int hours;
int minutes;
int seconds;

void showTime() {

cout << hours <<
min <<
secs;

Time(int hours, int minutes,) {
int seconds;

T~~ime~~.

Hours (t-hour),

minutes (t-minutes)

seconds (t-seconds)

2 4

3;

void main()

{

 Time Time1(12, 15, 15);

 Time1. ShowTime();

}

Q4 What is destructor? Why we need it?

A4 A destructor as name suggests which are if removed a constructor within delete.

Matrix:: ~Matrix()

{

 for (int i = 0; i < n; i++) {
 delete p[i];

}

class Test &
public:

Test() L.

count⁺ⁱ

y

~Test() L

count⁻ⁱ

y

Q5

Explain unary & binary
operator.

Ans

Unary

Let us consider the
unary minus operator i.e.
operator when used as a
unary, takes just one operand.

→ We know that this operator changes the sign of operand when applied to both data item.

→ The answer will be when applied to an object should change the sign of each of its data

Q) #include <iostream>
using namespace std;

Class Space 2

```
int x;  
int y;  
int z;
```

public :

```
void add Data (data  
int x,  
int y);
```

word opncl - C';

y;

at next) L

opncl - S

S, opncl a C¹⁰, = 10);
~~S.~~
~~S.~~

y

Q6

Explain need of friend as operator function in overriding an operator.

Ans As returning value, we can use friend as operator朋友

→ At first, it is a handle.

There are two cases where it is really important to use friend function.

→ Instead if we want to provide we can,

→ If we use friend, both cases can be understood

~~int exit()~~

Q8

What is template function?
Explain non generic parameter
in template function.

A

Function templates are
generic functions which work
for any data type that is
passed to them

- The data type is not specified while writing the function
- while using

Non-numerical parameters.

1. we can also pass non-numeric arguments to a function.
2. However pass, we can size of all to static functions shown in the following.

template < class T >
 |
 | < typename T >

NEW INDIA
PAGE NO.: DATE

Old compilers don't support this

Template function in C++

→ A template is simple, yet powerful tool.

→ The idea is to pass data type as parameter; so we don't need to write same code for different data types.

→ For instance, a company need to sort() different data types.

Rather than writing and maintaining multiple codes, if we can write sort() once and pass data type as parameter.

Syntax

template < typename T >

keyword "class" can also be used

These are expanded at compile time. They are like macros.

Difference is compiled does type checking before template expansion.

Source code contains only function/class, but compiled code may contain multiple copies of functions/classes.

template <type> max >

T mymax(T x, T y)

{
return (x > y) ? x : y;
}

int main ()
{

cout << "Enter x";
cin >> x;

cout << "Enter y";
cin >> y;

a = myMax < int > (3, 7);

b = myMax < char > ('q', 'e');
cout << a << b;
return 0;

}

Computer internally generates and adds below code.

```
int myMax(int x, int y)
{
    return (x>y) ? x : y;
}
```

```
char myMax(char x, char y)
{
    return (x>y) ? x : y;
}
```

Generic programming enables one to write general algos. which will work with all data types. It eliminates need to create different algos if data type is int, float or a character.

Function Templates



We write generic function that can be used for various data types.

Example:- sort(), max(), min(), printarray().



Class Templates

They are useful when a class defines something that is independent of data type.

Can be useful for classes like.

- linked list
- Binary Tree
- Stack
- Queue
- Array, etc...

Q3

COD

Explain static data member in class template and explain use of export keyword.

Ans

Export:

The export keyword is useful when following the second model of it where we define a template function at single place and its declaration at other place.

This is useful in instant we can find them useful for other application.

2. static data members in
class template.

Ans

The template class can also
have static data members.

→ The way a static member is
defined is analogous to way
member functions are defined
outside the template class.

Q1 Explain various types of C++
overloading.

A1 In C++, object-oriented,
objects represent classes in C++.

cout is an object of the
output streams class, while also
in an object of the input stream class,

2) In C++, it is possible to
have the program accept inputs
known as well defined manipulators
in C++.

Q12 Explain iOS min function
for formally.

a. width()

Ans It specifies the minimum field width for display.

The width() resizes itself after the first.

cont "Rolling":

b. precision()

It specifies precision, that is number of digits to be displayed after the decimal point.

(3) fill C)

The function fill the subsequent empty portions of field by the character

cont. fill ('+'):

cont. with (0) ;

cont C " Roll over";

(b) setf()

This specifies the font flags are as before right justification, select type notation

(c) unsetf()

This provides the undo operation in setf().

Q1)

what is manipulators.

A2)

manipulators are special function for formatting.

They can do all the formating that is done by the i/o's manip function.

setw()

Ans

It is set the width of field
cout << setw(5) << endl;

2)

setprecision()

Used to set the floating point precision after decimal points.

cout << setprecision(3) << endl;

(3)

setfill()

Used to fill empty obtain
after using any the
cout put.

cout << setfill('x') << str[0];

(4) endl :

Used to line in next line

cout << "Hello" << endl <<
"World".

Output:

Hello
world.

Name: Ajinkya
Rathod

Subject: OOP

Assignment 3

Q1 what is inheritance? Explain multiple inheritance and how to solve that?

Aⁿ The mechanism of deriving a new class from an old class is called inheritance.

A derived class with only one base class is called single inheritance and with several base classes is called multiple inheritance.

Multiple Inheritance

A) A class can inherit the attribute of two or more more classes shown in below figure.

→ Multiple inheritance allows us to combine the defining parts of different classes.

B-1

B-2

B-3

D

Q

Explain derivation using
differences method?

A:

+ Public derivation:

First case:

The first case where a class
class a derived using a public
access specific base like basic
class. The private member of
base class is copied by derived
class is not visible.

Base

Derived

private:

Pnt a;

private:

Pnt a;

public:

int ac;

public:

void bc;

Protected Privileges

In case of protection the protected members of base class become protected members at derived class.

Runtime Polymorphism

This type of poly is achieved by function overriding.

Function Overriding:

On other hand consider a new class has definition of one of member function. Then another class is said to be overridden.

Q5

Describe use of pointers.

Ans Pointer is address not refers

to another data variable by

Validly memory address rather than
data.

→ Declaring & Initializing pointer

datatype * pointer-variable;

Ex) ~~#include <iostream.h>~~
~~#include <conio.h>~~

int main()

int a=10;

* p = &a;

pdr = &a;

cout << value << * pdr;

* pdr = * pdr + 1;

y

Q6

Explain

(i) `fseek()`

It is used to move file pointer.

→ `SEEK - END` → End
`SEEK - SET` → Start
`SEEK - CUR` → Current

(ii) `f.eof()`

Syntax: int `f.eof`.

The `f.eof` function takes a file object as argument

and return an integer value

which specifies it end of file.

It is defined in `unistd.h`

(ii) `fread()`:

The `fread()` function is used to read a specific no. of characters from given input file.

Syntax: `size_t fread (void *buff,`
`size of size, file fptr)`

#include <fstream>
#include <iomanip>

using namespace std;

int main()

{

FILE *fp;

char buff[100];

fp = fopen("data.txt", "r");

```
while(!feof(fp)) {
```

```
    fread(buffer, sizeof(Button), 1, fp);
```

```
    cout << buffer;
```

```
}
```

```
fclose(fp);
```

```
3
```

(i) `fopen()`

The `fopen()` function opens a file indicated by

file as return a stream associated with that file.

If `r, wa, rt, we, -> Text`

`rb, wb, ab, wbt, g`

`Binary`

(V)

close

int close (file + psm)

The close() function takes a file object, a file stem which is to be closed. All data that are buffered but not written is flushed to the o.s. and all unread buffered data are discarded.

Q2

Explain Namespace in Java

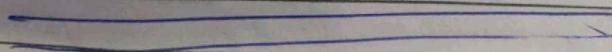
Ans

Namespaces can be used
used in various ways.

It's possible to have many
but,

→ `<!DOCTYPE html>` declaration about
which access specifier like
public, private, etc.

using namespace etc.



Q9 Advantages of saving a data in binary form:

Ans In binary files, numbers are stored in binary form,

It is stored in ms. D. Though characters are still stored in ASCII equivalents.

→ Binary file, a record is usually of the same file.

It also makes possible to provide file using a reader or a cat..

Q10 Explain BTL in detail?

→ STL stands for standard Template library is generic software components and algorithms by objects called containers.

→ STL components which are new part of the standard C++ library are defined in using namespace std.

* Component of STL

→ The STL contains several components but it covers all three key components. They are

- Containers
- Algorithm
- Iterator

* Container

A container is an object that actually stores data.

It stores data in organized manner. The STL containers are implemented by template classes and therefore customized to type of data.

* Algorithm

An algo is procedure that is used to process the data contained in memory. The STL includes many different kind of algs.

Iterator

An iterator is an object that points to an element in a container. We can use iterator to move through contents of containers.

We can increment or decrement them. Iterators connect algorithms with containers and play a key role in manipulations of data stored in containers.