MODEL QUESTION PAPER SET- 1 : 2021 - 22

MM : 50 **COMPUTER SCIENCE Paper – I (THEORY)** **Time : 3 Hrs**

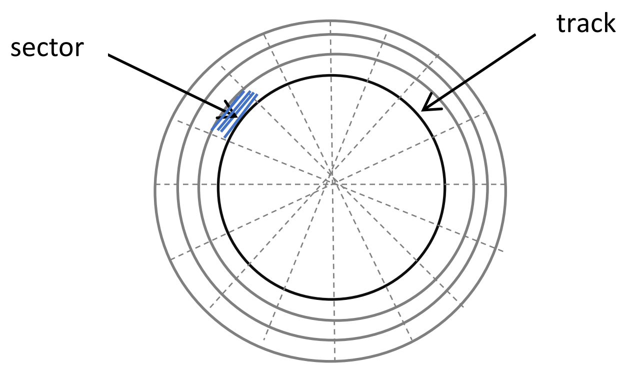
SOLUTIONS

Entire Syllabus

Q 1 (A)

i. (a) BR
ii. (c) double
iii. (b) Linux
iv. (b) Process Management

(B) i. Hard disk a storage device of a Computer with very high storage capacity.
It is made up of multiple magnetic discs.
Tracks : The magnetic discs are made up of multiple concentric circles. These circles store data.
One disc can have few hundred tracks.
Sector : Each track is divided into multiple parts of equal size.
These parts are called as sectors.



Head : Its an electronic part of a hard disk using which data is written on a disk and read from a disk.
Every disk has two sides. So for every disk there are two heads.
The heads are moved backward or forward to place them on particular track.

ii. Constructor :

1. Constructor is a member function of a class.
2. Its name is same as class name.
3. It works automatically when object of the class is created.
4. It can have zero or multiple parameters.
5. Constructor doesn’t have return type, not even void.
6. One can overload constructors.
7. Its defined as public member of a class.

iii. **1) Node :** One element in a collection is called as Node.
A node consists of two parts : Information and Links.
Information is data about any entity like Book, employee or student.
One node can have one or more links.

The links are used to link or refer to another element in a collection.

2) Record : Record holds information about one entity or one object.

It stores information in the form of multiple attributes of an entity and corresponding values.

Following figure shows Record of an employee

Emp Id	Name	salary	Designation	← attributes
2310	Ajay Patil	25000	Accountant	← values

3) File : File is a collection of records about particular type of entities e.g.

A file can be collection of records of multiple employees.

Emp Id	**Name**	**salary**	**Designation**
2310	Ajay Patil	25000	Accountant
2311	Rita Shah	40000	Jr Programmer

Q. 2 (A)

i. **Virus detection :** Virus is a program written with wrong intention. It gets attached to some binary file and perform some illegal operations.

For detecting virus we use Antivirus software.

The antivirus checks the integrity of a file. If any mismatch is found its treated as infected file.

In some cases an antivirus remains in memory and keep track of any suspicious behavior by any file. If such behavior is found its treated as infected by virus.

Thus Virus is detected.

Prevention : Data can’t be cured Once data is affected by Virus. So prevention is better.

Use legal and licensed software.

Run anti-virus frequently.

Take back-ups of data frequently.

ii. Polymorphism is a feature of OOP, which stands for ‘same name, multiple forms’.

Compile time polymorphism : Suppose there are multiple things e.g. functions present in a program.

When a program is compiled, the compiler makes a decision about which function to run at a particular function call, by checking the function parameters.

This is also called as early binding or static binding.

Function overloading, Operator overloading are examples of Compile time polymorphism.

Runtime Polymorphism : This occurs when a call to function is resolved at run time i.e. when a program is running.

Also, called as Late binding or dynamic binding.

This occurs in case of Virtual functions and inheritance.

iii. **Delete an element in array of size N:**

In the following algorithm A is an array, NO is an element to delete, N is number of element.

```
DELETE ( A, NO, N )
1.  SET  J = 1
```

```
2. WHILE  A[ J ]  <> NO REPEAT
      1.  J = J + 1
3. REPEAT FOR  K=J  TO  N-1
      1.  A[ K ]  = A[ K+1 ]
4. N = N-1
```

(B) i. Operator overloading:

Rules :

1. Only existing operators of C++ can be overloaded.
2. The syntax of the overloaded operator remains same as original operator.
3. The precedence level of overloaded operator remains same as that defined by C++.
4. While using overloaded operator one of the operands must be object type for which operator overloading is done.

Operator function : To overload a particular operator a special function called as operator function must be defined as a member of required class.

Syntax :

```
ReturnType   operator Symbol  ( Parameters )
{
// Statements
}
```

Here, operator is a keyword.

Symbol is an operator (e.g. +, != , > etc) to be overloaded.

ii. Scope resolution operator :

An operator   :: is called as scope resolution operator.

The operator is used for multiple purposes.

e.g.

Defining member function outside class.

Defining static data member outside class.

Accessing static members.

It’s also used to access global variables inside a function.

To access global variable we use syntax

```
:: VariableName
```

e.g.

```
int  a = 100;
void main()
{
int  a = 20;
…
cout <<  “local a=” <<  a << endl;
cout << “global a=” << :: a << endl;
}
```

Will output :

```
local a=20
global a=100
```

Q 3 (A)

i. Features of friend functions:

1. These functions are not member functions of a class.
2. These can access all member of a class: (private , protected and public) using class object.
3. Since they are not member of a class, they don't require object to call them.
4. We can define one function as friend of multiple classes.

ii. HR tag : The tag is a standalone tag.
It is used to display a horizontal line on a browser window.
Using color, size and width we can format the line.
By default line is black colour and occupies full width of browser window.
e.g. <HR color='red' width='50%' >

MARQUEE tag : The tag is used to display scrolling text or image on a browser window.
Using attributes like direction, behavior, height, scrolldelay etc one can display different types of scrolling messages or images.
e.g. <MARQUEE direction='right' behavior='slide'> This is demo text </MARQUEE>

PRE tag : The tag is used to display pre-formatted text on a browser.
All the new lines, tabs, spaces will be retained while displaying output.

```
<PRE>
1.  Pencil      10Rs
2.  Pen         50Rs
3.  Eraser      10Rs
</PRE>
```

Will show the text as stored in HTML file,

```
1.  Pencil      10Rs
2.  Pen         50Rs
3.  Eraser      10Rs
```

iii.

1. Windows NT is multitasking, multiuser and multithreading operating system.
2. A user will get faster response eventhough multiple applications are running.
3. Windows NT supports virtual memory management system to allow multiprogramming.
4. Symmetric multiprocessing in windows NT allows it to schedule various tasks on any C.P.U. in a multiprocessor system.
5. Windows NT is a 32-bit operating system.
6. Windows NT uses New Technology File Systems (NTFS), which implements fault tolerance, security and has support for very large files.

(B) i. Opening and closing file :

To open a file we create object of a required class.
Then we open a file using some open() function of the class.

Syntax : open(FileName, Mode);

In the above syntax,
File Name is a string that mentions full file name with its parent folder path.
If the program and the file to operate are in same folder, then only filename is enough.

e.g. To open file for reading,

```
ifstream  f;     // object of ifstream
        f.open( "mydata.txt" );   // opens the file  (default input mode)
```

We can also open a file *using constructor* of File IO classes,
e.g. open file for reading,

```
        ifstream  f ( "mydata.txt" );     // object of ifstream
```

Closing a file :
To close a file we use close() function of any File IO class.
Its common for all classes.
e.g.,

```
        f.close();     // close the file referred by f object
```

File open Modes :

ios::in	Open for input operations
ios::out	Open for output operations
ios::binary	Open for binary mode
ios::ate	Set the initial position at the end of the file

ii. Multiprocessing :
OS supports running multiple programs at time in a computer.
Every running program is called as a Process.
Thus, feature of OS that supports running multiple processes simultaneously is called as Multiprocessing.
This increases efficiency of system.
Process management unit of an OS manages and controls the multiprocessing.
For this the time of a CPU is shared between multiple processes.
This is achieved by Context switching.
Assume three processes A, B and C are running simultaneously.
Process A works for some amount of time and the status of CPU and process A is saved. Then process is loaded B starts and it works for amount of time. Then status of B is saved and C is loaded.
After C works for some time its status is saved and process A status is restored and it starts working.

Q.4. (A)

i. Static data members :
Static data members are defined with keyword 'static'.
Object is not required to access Static data members outside a class.
They are accessible outside the class using syntax : Classname::VaribaleName
Only one copy of static data members is created and shared by all the object of the class.
By default their value is zero.

ii. Destructors :
These are the member functions of a class with same name as class name preceded by symbol ~.
Destructors work automatically when object of class is destroyed i.e. when memory for object is deallocated.
This deallocation occurs when program terminates or a function ends.
Destructor can't have parameters.

Also, they don't have any data type.
Destructors are used to free the memory allocated by the object.

iii. Working of Stack data structure :

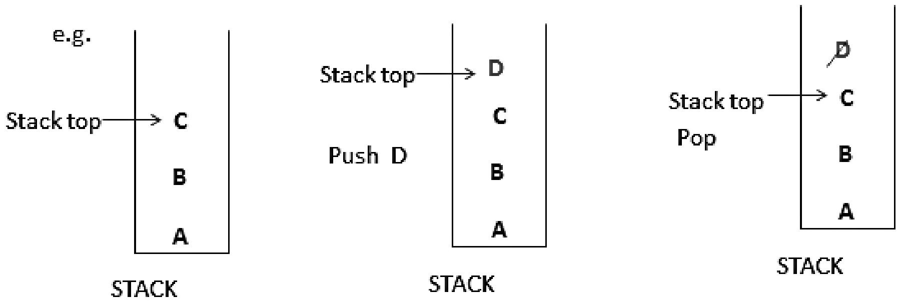
Stack:

- It's a linear data structure
- It is a List type structure.
- It works on 'Last in first out' (LIFO) principle.
- Data can be added or removed only from one end of the collection, called as stack top.

There are two main operations possible on stock:

Push : to add element at the end of the stack

Pop: to remove top element from a stack



(B)

i. **Virtual Memory :**

1) In multiprogramming systems the main memory (RAM) Is not sufficient to run all the processes simultaneously.
Sometimes the size of Process itself can be biggerthan the main memory size.
In such cases Virtual Memory technique is used by Memory Management unit of OS.

2) For this some part of Disk (e.g. Hard disk) in the system Is used as Memory.

3) It uses Paging or Segmentation or both the techniques..

4) When a program start working Some pages of program are loaded into Page frames according to availability.

While running a program if specific page of a program is not found in main memory. Page Fault occurs and that page is loaded from Disk.

(1) **Locality Reference :**

Its prediction about whether a particular page will be used in near future according to its past behaviour. If it may not be used in near future it may be removed from memory and stored on a disk.

(2) **Page fault :**

In a system when process is executing , not all the pages are loaded into memory.
If any instruction to be executed or any data to be accessed is in a page which is Not in the memory, then such situation is called as page fault. Here, the required page is loaded from disk into memory.

(3) **Working Set :**

The set of pages currently Active and used by process is called as working set.

ii. Priorities of processes :

Priority:

Concept of arranging ready processes in a Queue, so that they can be dispatched

1. **External priority:**
   Specified by user as the time of running program.
   Can be changed by user while process is running.
   Default priority.
2. **Internal priority:**
   Scheduling algorithms of OS.
   SJF (shortest job first) algo,
3. **Purchased priority:**
   User pays more to give higher priority to a process.
   OS keeps track.
4. **Time slice:**
   The CPU time is broken info equal time slots.
   and each process gets typical amount of time to execute/

Q.5 (A)

i. prog. Display 10 Fibonacci numbers.

```
#include<iostream.h>
 void main()
 {
 int   a = 0, b=1 , c , i ;
 cout<<"10 Fibonacci numbers\n";
 for ( i=1; i<=10; i++ )
 {
 cout<< a << endl;
 c = a + b ;
 a = b;
 b = c;
 }
 }
```

```
Output :
10 Fibonacci numbers
0
1
1
2
3
5
8
13
21
34
```

ii. prog. Class circle with constructor.

```
#include<iostream.h>
class  Circle
{
float  radius;
public :
Circle()
{
cout<< "Enter radius-" ;
cin >> radius;
}
void calculate()
{
float  area, circ;
area = 3.14 * radius * radius;
circ = 2 * 3.14 * radius;
cout<< "Area =" << area << endl ;
cout<< "Circumference =" << circ << endl ;
}
} ;
void main()
{
Circle  c;
c.calculate();
}
```

```
Output :

Enter radius-10

Area = 314.0

Circumference =62.8
```

iii. HTML page to display table:

```
<html>
<head> </head>
<body>
<table  width='40%'  border='1'>
<tr> <td  rowspan='2' align='center'> Year </td>  <td colspan='3' align= 'center'> Students </td>
</tr>
<tr> <td > Boys </td>  <td> Girls </td> <td> Total </td>
</tr>
<tr> <td > 2019 </td>  <td> 50 </td> <td> 48 </td><td> 98 </td>
</tr>
<tr> <td > 2020</td>  <td> 60 </td> <td> 63 </td><td> 123 </td>
</tr>
</table>
</body>
</html>
```

OR

Q.5.

i. Prog. Largest number in array :

```
#include<iostream.h>
void main()
{
int   a[10] , largest, i ;
cout<<"Enter 10 numbers\n";
for ( i=0; i< 10; i++ )
{
cin>> a[i];
}
largest  = a[0];
for ( i=0; i< 10; i++ )
{
if( a[i]  > largest )
largest = a[i] ;
}
cout<< "Largest number = " << largest;
}
```

```
Output :
Enter 10 numbers
20
30
15
40
25
17
38
50
35
42
Largest number = 50
```

ii.

```
#include<iostream.h>
class  Student
{
public :
int  rollno;
char  name[40];
void  input_student()
{
cout<< “Enter Name, Roll no\n” ;
cin.getline( name, 40) ;
cin>>rollno;
}
} ;
class  Marks : public Student
{
public :
int  phy, chem, maths;
void input_marks()
{
cout<< “Enter three marks\n” ;
cin>> phy >> chem>> maths ;
}
} ;
class  Result : public Marks
{
public :
void  output_result()
{
flaot  total, avg;
total = phy + chem + maths;
avg = total / 3;
cout<< “Physics=”<< phy <<endl ;
cout<< “Chemistry=”<< chem <<endl ;
cout<< “Maths=”<< maths <<endl ;
cout<< “Total=”<< total <<endl ;
cout<< “Average=”<< avg <<endl ;
}
} ;
void main()
{
Result  r;
r.input_student();
r.input_marks();
r.output_result()
}
```

```
Output :
Enter radius-10
Area = 314.0
Circumference =62.8
```

iii. HTML Web page :

```
<html>
<head> </head>
<body>
<h1>XYZ College </h1>
<hr>
<font size='+1' color='red'> Department </font> <br>
<ul>
<li> Arts </li>
<li>Science</li>
<li>Commerce</li>
</ul>
<br>
To view College campus  <a  href='d:\images\campus.html'> Click here  </a>
</body>
</html>
```
