Rao IIT Academy
Symbol of Excellence and Perfection
JEE | MEDICAL-UG | BOARDS | KVPY | NTSE | OLYMPIADS

XII HSC - BOARD - MARCH - 2016

Date: 11.03.2016

COMPUTER SCIENCE - I (D-9)

SOLUTIONS

1. **(A)**

(a) (iii) <TABLE> **[1 Mark]**

(b) (iv) 8 **[1 Mark]**

(c) (ii) UNIX **[1 Mark]**

(d) (i) Process **[1 Mark]**

1. **(B)**

(a) (1) In computer ,user can't interact directly to hard disc because In the hard disc,the computer language is in the form of binary form i.e 0's and 1's.

(2) This binary form is a machine level language.

(3) Computer doesn't understands human level language.

(4) For Interaction with hard disc, the ASCII language is required ,which system only knows,and not human being. **[3 Marks]**

Topic:C++ programming;XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(b) (1) The constructor name is always same as the class name.

(2) They do not have return types, not even void and therefore, they cannot return values.

(3) They cannot be static or virtual.

(4) They should be declared in public section.

(5) They cannot be inherited though a derived class can call base class constructor.

(6) Like other C++ functions, they can have default arguments.

(7) We cannot refer to their address.

(8) An object with a constructor cannot be used as a member of union.

(9) They make implicit calls to the operators 'new' and 'delete' when memory allocation is required.

(10) When a constructor is declared for a class, initialization of class objects become mandatory, since constructor is invoked automatically when the objects are created. **[3 Marks]**

Topic:C++ programming;Sub-Topic:Constructor; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(c) (i) **File:** It is the collection of records of the entities in a given set.

e.g.

	Roll No.	Serial Number	Name	Address	Telelphone
Record →	1	s123	Sachin	Pune	25467890
	2	s124	Mita	Mumbai	23456789

↓

Field

(ii) **Record:** Record is a collection of field values of a given entity set.

(iii) **Key-Field:-** Field is a single elementary unit of information representing an attribute of an entity.

[3 Marks]

Topic:Data Structure;Sub-Topic:Data structure; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

2. **(A)**

(a) **Explanation:**

(1) Virus detection

(i) Normally virus detection program checks integrity of binary files. It maintains a checksum on each file. At regular frequency detection program calculates checksum and matches with original one. If there is mismatch then that program may be infected.

(ii) Some programs reside in the memory and continuously monitor memory and I/O operations against virus.

(2) Virus removal

There are some viruses whose bit pattern in the code can be predicted. The virus removal program scans the disk for the patterns of known viruses and on detection it removes them.

(3) Virus prevention

For prevention of virus the user can take the following precautions

(i) Always buy legal copies of software.

(ii) Take frequent backups of data

(iii) Run monitor programs frequently to detect virus. **[3 Marks]**

Topic:Operating System;Sub-Topic:Virus; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(b) Polymorphism refers to identically named methods (member functions) that have different behavour depending on the type of object they refer.

Polymorphism simply means “one name, multiple forms.

Runtime polymorphism :

(1) In some situations, it is nice to select appropriate member function to be invoked while the program is running. This is known as runtime polymorphism.

(2) The types of polymorphisms and their examples are shown in following figure:

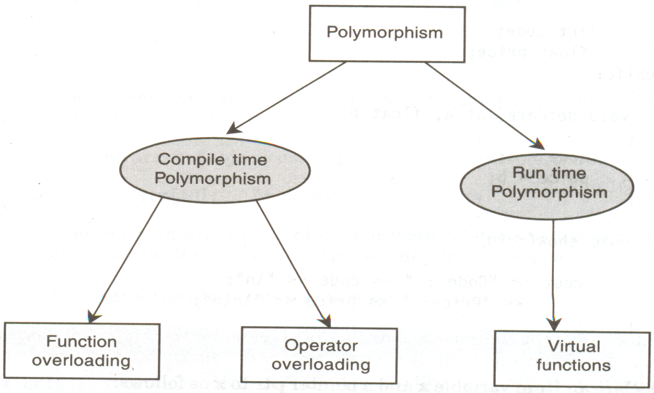


Fig. 9.1 Achieving polymorphism

Here, show ( ) function is used to print values of object of both the classes A and B. The prototype of show ( ) is the same in both the places, the function is not overloaded and therefore static binding does not apply.

(3) In such situations, the appropriate member function can be selected at runtime and it is known as runtime polymorphism.

Compile Time Polymorphism :

(1) Function overloading and operator overloading are the examples of compile time polymerphism.

(2) In this case, the overloaded member functions are selected for invoking by matching arugments, both type and number.

(3) This information is known to the compiler at the compile time and, therefore the compiler is able to select the appropriate function for a particular call at the compile time itself. This is known as compile time polymorphism.

(4) Compile time polymorphism is also called as early binding or static binding or static linking. Early binding simply means that an object is bound to its function at compile time. **[3 Marks]**

Topic:C++ programming;Sub-Topic:Polymorphism; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(c) (1) Inserting element to an array is the process of adding an element to the existing elements of array.

(2) The element can be easily inserted at the end of an array. But for insertion in the middle of an array it is required to move the elements one byte forward.

(3) The following algorithm inserts a data element in an array:-

Algorithm for inserting element into a linear array

```
INSERT [LA, N, K, ITEM]
LA = Linear array
N = Total no. of elements in the array
K = Any positive integer, K<=N
This algorithm inserts an element ITEM into the K^th position in LA.
1.[Initialize Counter]                Set J: = N
2.Repeat steps 3 and 4 while J ≥ K
3.[Move J^th element downward]  Set LA [J+1]: = LA [J]
4.[Decrease Counter]                  Set J: = J-1
  [End of step 2 loop]
5.[Insert element]  Set LA [K]: = ITEM
6.[Reset N]                           Set N: = N+1
7.Exit
```

Number	9 is inserted	4 is deleted
3	3	3
4	4	9
5	9	5
6	5	6
8	6	8
	8	
Original Array	Addition of element in middle array	Deletion of element from array

[3 Marks]

Topic:Data Structure;Sub-Topic:Array; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

2. (B)

(a) (1) To define an additional task to an operator, it specify what it means in relation to the class to which the operator is applied. This is done with the help of a special function, called operator function, which describes the task.

(2) In short, a function which defines additional task to an operator or which gives a special meaning to an operator is called the operator function.

(3) The general form of operator function is,

```
return-type class-name : : operato op(argument list)
{
   function body//task defined
}
```

Where return type is the of value returned by the specified operation and op is the operator being overloaded.

The op is preceded by the keyword operator. **Operator op is** the function name.

(4) Operator functions must be either member functions (non-static) or friend functions.

(5) The basic difference between operator function as a friend function and as member function is that a friend function will have only one argument for unary operators and only one for binary operators. This is because the object used to invoke the member function is passed implicitly and therefore is available for the member function. This is not the case with friend function. Arguments may be passed either by value or by reference. **[4 Marks]**

Topic:C++ programming;Sub-Topic:Operator Overloading and type conversion; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(b) (1) Local data or variable is that data which comes inside the class.

(2) Global data or variable that data which comes outside the class.

(3) So, For accesing global data ,compiler requires scope resolution operator.

(4) While in local data,scope resolution is not required.

(5) example:

```
 int a ;          //global data
void main()
{
   int a;   // local data
}
```

[4 Marks]

Topic:C++ programming;Sub-Topic:Data; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

3. (A)

(a) (1) It is not in the scope of the class to which it has been declared as friend.

(2) Since it is not in the scope of a class, it cannot be called using the object of that class.

(3) It can be invoked like a normal function without the help of any object.

(4) Unlike member functions, it cannot access the member names directly and has to use an object name and dot membership operator with each member name.

(5) It can be declared either in the public or the private part of a class without affecting its meaning.

(6) It has the objects as arguments. **[3 Marks]**

Topic:C++ programming;Sub-Topic:Friend function; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(b) <BR> tag

(1) <BR> tag insert line break into a text flow.

(2) It tells the browser to wrap the text that follows onto a new line without inserting any extra space between the lines.

<HR> tag

(1) A web page can be divided into separate sections by using horizontal rulers, <HR> tag. It is also called as horizontal line.

(2) This tag is mostly used for decorative purposes.

<IMG> tag

(1) <IMG> is image tag. Its purpose is to include graphic images in the body of the web page.

(2) To make an image as a separate paragraph, it is enclosed within paragraph elements.

e.g. <P> <IMG SRC = "C : \ My Documents \ Lion.jpg"> </P> **[3 Marks]**

Topic:HTML;Sub-Topic:Tags; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(c) The features of Windows NT are as follows:-

(1) It is multitasking, multi-user and multithreading operating system.

(2) It also supports virtual memory management system to allow multiprogramming.

(3) Symmetric multiprocessing allows to it to schedule various tasks on any CPU in a multiprocessor system.

(4) It uses New Technology File System which implements fault tolerance, security and support for large files.

(5) It is 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. **[3 Marks]**

Topic:Operating System;Sub-Topic:Windows NT; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

3. (B)

(a) The classes in C++ for file stream operation are :

a. filebuf

b. fstreambase

c. ifstream

d. ofstream

e. fstream

The file can be opened with two ways:

1. Using the constructor function of the class and

2. Using the member function open() of the class.

Using Constructor:

Ofstream outfile("sample.txt"); // output only

Ifstream infile("sample.txt"); // input only

Using open( ):

Ofstream outfile; // create stream

Outfile.open("Sample.txt"); // connect stream to sample.txt

The files can be closed using close() function.

Outfile.close( ); // disconnect stream from file buffer

File modes:

App : append to the end-of-file

Ate : Go to end-of-file on opening

Binary : Binary file

In : Open file for reading only.

Out : open file for write only. **[4 Marks]**

Topic:C++ programming_;Sub-File modes; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

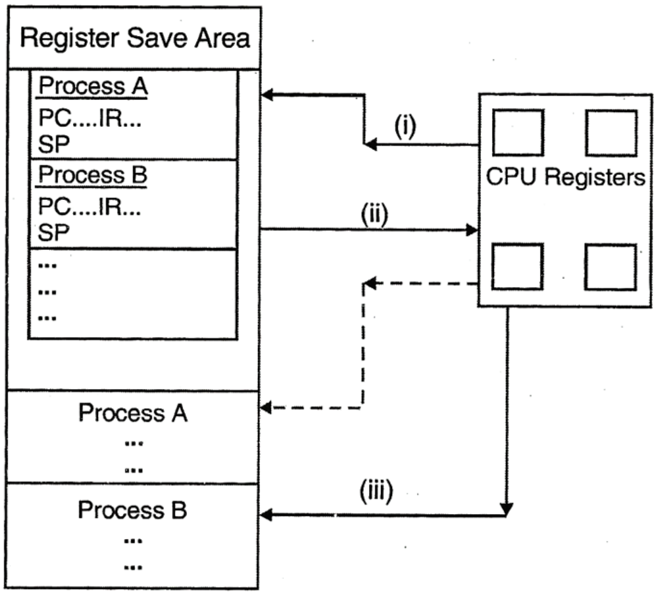
(b) (1) The number of processes running simultaneously competing for CPU is called multiprogramming.

(2) Multiprogramming is used for increasing the CPU utilization and reducing the idleness of CPU.

(3) In multiprogramming CPU can execute multiple processes at a time.

e.g. When process1 waits for an external event (I/O operation), the processor executes process 2.

(4) The switching time between two processes is called as context switching.



Context switching (From Process A to B)

(5) Context means storing the status of process. In that memory contents and registers like Program Counter, Instruction Register, Accumulator, general purpose registers are stored. Each process has a register save area and in that area context for the process is stored.

(6) Suppose Process A is under execution and it requires I/O operations, then process B should start. So there would be context switching from A to B. The context of process A is stored and restores the process B status which is already in memory. When process A is scheduled again, then it is restored in CPU and continues from the leftover point. **[3 Marks]**

Topic:Operating System;Sub-Topic:Multiprogramming; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

4. **(A)**

(a) The special characteristics of a static data member in a class are as follows:

(1) It is initialized to zero when the first object of its class is created. No other initialization is permitted.

(2) Only one copy of that member is created for the entire class and is shared by all the objects of that class, no matter how many objects are created.

(3) It is visible only within the class, but its life time is the entire program.

(4) Static members of a class are not specific to an instance of this class.

(5) The class constructor does not initialize static variables.

(6) Thus, Instead of object, with the help of the class the static members are called. **[3 Marks]**

Topic:C++ programming;Sub-Topic:Static members; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(b) **Use of memory management operator :** new **and** delete

(1) The **new** operator is used to create objects of any type.

(2) The syntax is as follows:

Pointer_variable = new data type;

(3) Example : p = new int; // p is pointer of type int .

(4) The **delete** operator is used to destroy the variable from the memory space so that the same space can be used for another use.

(5) The syntax of delete is as follows:

delete pointer variable;

(6) Example: delete p; // delete pointer variable p from memory. **[3 Marks]**

Topic:C++ Programming;Sub-Topic:Memory Management Operators; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(c) **Record:-**

A record is collection of related data items. Each data item is termed as field. File is collection of similar records. Each data item may be a group item composed of sub items.

(1) Records are a collection of nonhomogeneous data so it can not be stored in an array.

(2) In entire file of records all data elements belonging to the same identifier will be of same type. So file may be stored in memory as collection of arrays.

(3) For representation we can use one array for each data item. All the arrays should be parallel.

(4) **Example:**

Student File:

Name	Address	Phone
Parth	12, J.K.Road	5667234
Shriniket	24, M.G. Road	5667890
Yash	12, Ramnagar	5467891

The following figure shows the representation of above file in three parallel arrays Name, Address, Phone.

Name		Address		Phone
Parth		12, J.K.Road		5667234
Shriniket		24, M.G. Road		5667890
Yash		12, Ramnagar		5467891

[3 Marks]

Topic:C++ Programming;Sub-Topic:Memory Management Operators; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

4. **(B)**

(a) **Virtual Memory :** Virtual memory is an attempt, which makes the execution of the processes possible, which may completely not in the main memory. Some part of the process may be on disk.

(1) **Locality of reference:** Clustering of page references in a certain time zone is called the principle of locality of reference. It gives some basis to forecast whether a page is likely to be referenced in the near future based on its passed behavior and so you can swap out the page.

(2) **Page Fault:** During the process execution, only few pages are in the memory. At that time an instruction is encountered which refers to any instruction or data in that page which is not in memory. So the page can not be referred and it is called as the page fault.

(3) **Working Set:** The set of pages in the physical memory actively referred to at any moment is called working set.

(4) **Page replacement policy:** As no. of processes and no. of pages in main memory for each process increase, all the page frames become occupied at some point of time. For the new page, the operating system has to overwrite some existing page in memory. The page to be replaced is selected by page replacement policy.

(5) **Dirty Page:** Before overwriting a page, the operating system has to check whether the page has to modified or not. If the page is modified, it becomes a dirty page. The dirty bit is used for checking whether the page is dirty or not.

(6) **Demand Paging:** A page is swapped in when it is demanded. This policy is known as demand paging. **[4 Marks]**

Topic:Operating system;Sub-Topic:Virtual Memory; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(b) (i) **Internal priority:** It is decided by scheduling algorithms. Their calculations are based on the current state of the process. The following scheduling algorithms are used for that.

(a) **Shortest Job First Algorithm:** The job whose expected time for completion is less is executed first.

(b) **Expected remaining time to complete:** It is same as SJF at the beginning but as the process progresses the time will change. At regular intervals operating system calculates the expected remaining time for completion and accordingly priority is determined.

(ii) **External Priority:** - The user specifies the priority externally at the time of initiating the process. If the user does not specify any priority, operating system assumes a default priority for that.

If the job is too urgent, system manager permits the process at a higher priority.

(iii) **Purchase Priority**: In this priority higher priority processes are charged at a higher rate. The operating system keeps the track of the time used by each process and charges it accordingly.

(iv) **Time Slice**: Each process is given a certain time for execution. The predetermined time given is called as time slice of the process. **[4 Marks]**

Topic:Operating system;Sub-Topic:Priority; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

5.

(a) //Program to generate 20 terms of fibbonacci series **[5 marks]**

```
#include <iostream.h>
void main ( )
{
        int f0, f1, f, n;
        f0 = 0;
        f1 = 1;
```

```
                cout <<"Fibbonacci Series :" ;
                cout << f0 ;
                cout << f1 ;
                        for (n = 2; n < 20; n++)
                        {
                                f = f0 + f1;
                                cout << f ;
                                f0 = f1;
                                f1 = f;
                        }
        }
```

[5 marks]

Topic:C++ programming;Sub-Topic:Program; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(b)

```
#include<iostream.h>
class temperature
{
        float cel;
        float far;
public : Temperature ( ) ; //constructor
        void convert ( );
        void print ( );
};
temperature : : temperature ( )
{
  cout<<"Enter the degree celsius";
  cin>>cel;
}
void temperature : : convert ( )
{
  far = cel * 9/5 + 32;
}
void temperature : : print ( )
{
 cout<<"The degree fahernheit is:";
 cout<<far;
}
void main ( )
{
 temperaute obj;
 obj.convert ( );
 obj.print ( );
}
```

[5 marks]

Topic:C++ Programming;Sub-Topic:Programe; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(c)

```
< HTML >
< BODY >
< TABLE border = “1” >
< Caption align = “ bottom” > Record < / caption >
< TR >
< TH rowspan = 2 > Year
< TH colspan = 3 > Students
< /TR >
< TR >
< TD > Boys
< TD > Girls
< TD > Total
< /TR >
< TR >
< TD > 2004
< TD > 25
< TD > 30
< TD > 55
< /TR >
< TR >
< TD > 2005
< TD > 80
< TD > 25
< TD > 105
< /TR >
< /TABLE >
< /BODY >
< /HTML >
```

[5 marks]

Topic:HTML; Sub-topic: Program_XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

OR

(a)

```
# include <iostream.h>
class average
{
float a, b, c, avg;
public : average ( ); //constructor
        void calculate ( );
        void print ( );
};
average : : average ( )
{
 cout<<“Enter numbers”;
  cin>> a >> b >> c;
}
void average : : calculate ( )
{
 avg = a + b + c/3;
}
void average : : print ( )
{
 cout <<“ The average of 3 nos is:” <<avg;
}
void main ( )
{
 average obj;
 obj calculate ( );
 obj print ( );
}
```

[5 marks]

Topic:C++ programming; Sub-topic: Program_XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(b)

```
# include <iostream.h>
 class student
 {
   protected :
       int roll_number ;
   public :
       void get_number (int) ;
       void put_number (void) ;
 } ;
 void student  : : get_number (int a)
 { roll_number=a ;}
 void student  : : put_number ( )
 { cout<<“roll number :”<<roll_number<<” \n” ; }
 class test :public student
 {
   protected :
       float sub1 ;
       float sub2 ;
       float sub3 ;
   public :
       void get_marks (float , float, float) ;
       void put_marks (void) ;
 } ;
 void test : : get_marks (float x, float y, float z)
 {  sub1=x; sub2=y; sub3=z;
 void test : : put_marks ( )
 {
   cout<<“marks in sub1=”<<sub1<<” \n” ;
   cout<<“marks in sub2=”<<sub2<<”\n” ;
```

```
    cout<<"marks in sub3="<<sub3<<" \n" ;
}
class result : public test
{
    float total ;
    public :
    void display (void) ;
} ;
void result : : display (void)
{
    total=sub1+sub2+sub3 ;
    put_number ( ) ;
    put_marks ( ) ;
    cout<<"total="<<total<<"\n" ;
 }
main ( )
{
    result student1;
    student1 . get_number (101) ;
    student1 . get_marks (75 . 0, 69 . 5, 55 . 5) ;
    student1 . display ( ) ;
}
```

[5 marks]

Topic:C++ Programming;Sub-Topic:Program; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam

(c)

```
<HTML>
<BODY>
<TABLE>
<TR> <CENTER> <H1> COLLEGE </H1> </CENTER> </TR> <BR>
<TR> Principal </TR>
<UL type= "circle">
    <LI> Vice Principal </LI>
   <LI> Professors </LI>
   <LI> Non teaching Staff </LI>
</UL>
For more details <A HREF = "C:\My Document \A1.HTML"> click here </A>
</TABLE>
</BODY>
</HTML>
```

[5 marks]

Topic:HTML;Sub-Topic:Program; XII–HSC Board Exam _Target-2016_Comp. Sci. I_NKcs Mam