1. OBJECT ORIENTED CONCEPTS USING C++

TWO MARK QUESTIONS: -

1. What is Object? **[Mar-07, June-08, Oct-09]**
   - A group of data and the operations are termed as object.
   - The operations represent the behavior of the object.

2. What is Encapsulation? **[Oct-07, 08, Mar-08,10]**
   The mechanism by which the data and functions are bound together within an object definition is called as encapsulation.

3. What is Polymorphism? **[Mar-09]**
   The ability of an object to respond differently to different messages is called as polymorphism.

4. What is meant by Inheritance? **[June-07]**
   The process of acquiring the Base class properties is called Inheritance.

5. What is the significance of an object?
   - An object is a group of related functions and data that serves those functions.
   - An object is a kind of self-sufficient “subprogram” with a specific functional area.

6. List any two advantages of Object Oriented Programming? **[June-09]**
   - Class data type allows programs to organize as objects that contain both data and functions
   - Data hiding or Abstraction of data provides security to data, as unrelated member functions cannot access its data.
   - Polymorphism reduces software complexity, as multiple definitions are permitted to an operator or function
   - Inheritance allows a class to be derived from an existing class, thus promoting reusability of code.

7. How is polymorphism different from inheritance?
   - Polymorphism reduces software complexity, as multiple definitions are permitted to an operator or function
   - Inheritance allows a class to be derived from an existing class, thus promoting reusability of code.

2. OVERVIEW OF C++

TWO MARK QUESTIONS: -

1. Define Tokens? **[Oct-09]**
   - The basic types are collectively called as Tokens
   - A token is the smallest individual unit in a program.

2. What are the classifications of Tokens? **[June-08]**
   Tokens are classified as Keywords, Identifiers, Constants, Operators and Punctuators

3. What are keywords? Give examples? **[Mar-09]**
   - Keywords have special meaning to the language compiler.
   - These are reserved words for special purpose
   - These words cannot be used as normal identifiers.

   Examples:- if, else, for, do, while, switch, case, break

4. Write a note on String Literal? **[June-09]**
   - It is a sequence of characters by double quotes.
   - These are treated as array of characters.
   - Each string literal is by default with special character '\0' which marks the end of a string. **Ex: -** "Testing"

5. List out the Relational operators with C++? **[Mar-08]**
   - Relational operators are used to compare numeric values.

   The relational operators are
   = = →equal to >→greater than <→lesser than
   >=→greater than or equal <=→lesser than or equal
   != not equal to

6. Write the conditional operator with example? **[Mar-07, Oct-07]**
   - A ternary operator (? :) is also called as conditional operator.
   - **General Syntax: -** E1?E2:E3 where E1,E2,E3 are operands
   - **Example: -** x=(a>b)?"True":" False"

7. Write a note on assignment operators? **[June-07]**
   - = is the simple assignment operator.
   - It is used to assign the result of an expression (on the right hand side) to the variable (on the left hand side of the operator).
   - **Ex:-** int a=10;

8. What are the two important purposes of void type?
   - To indicate that a function does not return a value
   - To declare a generic pointer

9. Give two uses of void data type? **[Mar-07]**
   - It indicates the compiler that the function does not return a value
   - It indicates that it holds nothing

10. What is the use of the operators related to pointer variable? **[June-07]**
    What are pointer variables? **[June-08]**
    - Pointer variables can store the address of other variables.
    - But the address stored in pointer variables should be of the same data type a pointer variable is pointing to.
    - **Ex:-** The asterisk (*) is used to declare the pointer variable. It is used to display the contents stored at a location. It is an unary operator.

11. How are the pointer variables declared? **[Oct-08]**
    int * iptr;
    Where int→ indicates that the pointer will point to an int data type
    * →instructs the compiler that the variable is pointer
    iptr → Name of the pointer variable

12. Write about User Defined Data Type? **[Oct-08]**
    - User Defined Data Type enables a programmer to invent own data type and define values it can assume.
    - This helps in improving readability of the program.

13. Write about the impact of modifiers in C++? **[Mar-07, 09, 10]**
    - unsigned modifies the range of the integer values as the sign bit is also used to store data.
    - Long increase the bytes for a particular data type, thus increasing the range of values

14. Write a note on enumerated data type? **[Oct-07]**
    Enumerated data type helps users in creating a list of identifiers, also called symbolic numeric constants of the type int.

15. Give the syntax and examples of enum data type? **[Oct-07]**
    Syntax: - enum data type identifier (value1, value2,.);
    Example: - enum holidays (Sunday, Saturday)

16. Write about Type definition and syntax in C++? **[Mar-08,10, Oct-08]**
    Users can define a variable that would represent an existing data type. It allows users to define such user defined data type identifier.
    Syntax: - typedef data_type user_defined_data_type_identifier
    Example:- type def int marks;

17. What is type cast? **[Oct-09]**

It refers to the process of changing the data type of the value stored in a variable.

18. What are the four storage specifiers in C++? **[Oct-07, June-09]**

The four storage specifiers are auto, static, extern and register

19. List out user defined data types?**[June-07]**

1. Structure 2. Union 3. Class and enumeration

20. Define size of operator in C++?

Size of is an operator. It returns the size (memory requirement) in terms of bytes, of the given expression or data type.

3. BASIC STATEMENTS

TWO MARK QUESTIONS: -

1. What are the different statements in C++?

Input/output, Declaration, Assignment, Control structures, Function call Object, Return.

2. What is the use of cin object?

- It is a standard input stream.
- Input stream represent the flow of data from the input device-Keyboard.
- It is available in a header file as isstream.h

3. What is the use of cout object?

- It is a standard output stream.
- Output stream normally flows to the screen display
- It is available in a header file as ostream.h

4. What are the various sections in C++ program? **[June-09]**

- Include files
- Declaration of variables, data type, user defined functions
- main( ) function

5. What do you know about assignment statements in C++ **[Mar-08]**

- An assignment statement, assigns value on the right hand side of an expression to the variable on the left hand side of the assignment operator
- = is the assignment operator. **Ex:-** num=10;

6. What are the control structures? What are the two main categories of control structures? **[Oct-07]**

- Program statements that cause jump of control from one part of a program to another is called as control statements or control structures.
- The two main categories of control structures are Decision making statements and looping statements.

7. Write the syntax of nested if statement? **[Oct-08]**

The statement sequence of if or else may contain another if statement ie., the if else statements can be nested within one another.

Syntax:- (To see PageNo.66)

8. What is the purpose of break statement?

- Break statement would exit the current loop only.
- It accomplishes jump from the current loop

9. What is the purpose of using main () function?

- When the program is executed the main () function will be automatically executed.
- It is from this block, that one needs to give call statements to the various modules that need to be executed and the other executable statements.

10. What is the purpose of continue statement? **[June-07]**

The continue statement forces the next iteration of the loop to take place, skipping any code following the continue statement in the loop body.

11. Write the rules for the formation of nested loops? **[June-08, Mar-09]**

- An outer loop and inner loop cannot have the same control variable, as it will lead to logical errors
- The inner loop must be completely nested inside the body of the outer loop

12. How is a pointer variable different from ordinary variable? **[Mar-08]**

Variable	Pointer
1. Variables are user defined name entities of memory locations that can store data	A pointer is a variable that hold a memory address
2. Special characters are not allowed to declare a variable	* is used to declare a pointer variable

FIVE MARK QUESTIONS: -

1. Explain switch statement with a suitable example?**[Oct-08,Mar-09,10]**

It is a multiple branching statement where based on a condition, the control is transferred to one of the many possible points.

Syntax: -

```
switch(expression)
{
case 1: action block1; break;
case 2: action block2; break;
case 3: action block3; break;
default: action block; break;
}
```

Example: -

```
#include <iostream.h>
#include <conio.h>
main()
{
int no;
clrscr();
cout<<"Enter a Number\n";
cin>>no;
switch(no)
{
case 1:cout<<" Number is One";break;
case 2:cout<<" Number is Two";break;
case 3:cout<<" Number is Three";break;
case 4:cout<<" Number is Four";break;
case 5:cout<<" Number is Five";break;
default: cout<<" Invalid Number";break;
}
getch();
}
```

Every action block should be terminated with a break statement.

Break statement: -

- Break statement would exit the current loop only.
- It accomplishes jump from the current loop

2. Explain Entry-check loop with example**? [March-2009,2010]**

While loop is called as the entry-check loop.

Syntax:-

```
while (condition)
{
  action block
}
```

- The body of the while loop will be executed only if the condition is true.
- The control exits the loop once the condition is evaluated to false.

Example:-

// To print the square of numbers between 2 to 5

```
#include<iostream.h>
#include<conio.h>
main()
{
int num=2;
clrscr();
while(num<6)
{
cout<<num*num<<'\t';
num+=1;   }
getch();  }
```

Output:-

4 9 16 25

Working of loop:-

1. Initializes the control variable num=2
2. Num<2 is evaluated, control is transferred to step 3, if the condition is TRUE
3. Print the square of the value stored in num
4. Increment num by 1
5. Control is transferred to step 2
6. End

3. Explain Exit-check loop with example? **[June-2007], [June-2009]**

- do….while is called as exit-check loop.
- The condition marks the last statement of the body of the loop.

Syntax:-

```
do
{
 action block
} while (condition);
```

Example:-

```
// To print the square of numbers between 2 to 5
#include<iostream.h>
#include<conio.h>
main()
{
int num=2;
clrscr();
do
{
cout<<num*num<<'\t';
num+=1;
} while(num<6);
getch();
}
```

Output:-

4 9 16 25

Working of loop:-

1. Initializes the control variable num=2
2. Num<2 is evaluated, control is transferred to step 3, if the condition is TRUE
3. Print the square of the value stored in num
4. Increment num by 1
5. Control is transferred to step 2
6. End

4. Explain Entry-controlled loop with example? **[June-2008]**
   - for loop is an entry controlled loop
   - It is used when an action is to be repeated for a predetermined number of times.

Syntax:-

```
for(initial value; test-condition; increment)
{
  action block
}
```

Example:-

```
for(i=1;i<=6;i++)
```

Increment
Test condition
Initialisation

Working of loop:-

The control variable is initialized first.
Test condition is evaluated.
The body of the loop is executed only if the condition is true.
The control variable is incremented and the test condition will be evaluated before the body of the loop is executed.
The loop is terminated when the test condition is false

Example:-

```
#include <iostream.h>
#include <conio.h>
main ()
{
int i,n,a=-1,b=1,c;
clrscr();
cout<<"How Many Terms\n";
cin>>n;
```

```
cout<<"Fibonacci Series\n";
for(i=1;i<=n;i++)
{
c=a+b;
a=b;
b=c;
cout<<c<<"\n";
}
getch();
}
```

5. Explain the different kinds of loops with example? **[March-2007]**
   (Refer the Q.No 2,3,4)

6. Explain the 'Nested.. if' statement with an example in C++? **[Oct-07,09]**
   The statement sequence of if or else may contain another if statement ie., the if else statements can be nested within one another.
   Syntax:- (To see PageNo.66) **Example:-** (To see PageNo.67)

4. FUNCTIONS

TWO MARK QUESTIONS: -

1. What are functions?
   - Functions are the building blocks of C++ programs.
   - It is also the executable segments in a program.
   - The starting points for the execution of a program is main( ).

2.What are the advantages of using functions in C++? **[June-07, Oct-07,09]**
   - Reduce the size of the program
   - Induce reusability of code
   - A function can be shared by other programs by compiling it separately and loading them together.

3. What is the main purpose of function prototype? **[Mar-09]**
   - It is help the compiler to check the data requirement of the function.
   - With function prototyping, a template is always used when declaring and defining a function.
   - When a function is called, the compiler uses the template to ensure that proper arguments are passed, and the return value is treated correctly.

4. What are the two methods used in Functions?
   1. Call by value method 2. Call by reference method

5. Difference between call by value and call by reference? **[June-09]**

Call by value: -

- The flow of data is always from the call statement to the function definition.
- Any change in the formal parameter is not reflected back to the actual parameter.

Call by reference: -

- Formal and actual parameters in reference type point to the same storage area.
- Any change in the formal parameter is reflected in actual parameter.

6. What are the rules for actual parameters?

It can be passed in the form of constants or variables or expressions to the formal parameters, which are of value type.

7. What is meant by actual parameters and formal parameters?

The parameter associated with call statement is called actual parameters and the parameter associated with function header is called formal parameters.

8. Write a note on inline functions? **[Mar-07,08, June-08, Oct-08]**

When the functions are small, the compiler replaces the function call statement by its definition ie, its code during program execution. This feature is called as inline function.

- An inline looks like a normal function in the source file but inserts the function's code directly into the calling program.
- Inline functions execute faster but require more memory space.

9. What is the use of scope resolution operator?

- :: is the scope resolution operator.
- It is used to refer variables declared at file level.
- This is helpful only under situations where the local and file scope variables have the same name.

FIVE MARK QUESTIONS: -

1. Define scope? Explain the different types of scopes variables in C++? **[June-07, Oct-08, Mar-09]**

Scope refers to the accessibility of a variable. There are four types of scopes in C++. They are

1. Local scope 2. Function scope 3. File scope 4. Class scope

1. Local scope: -

- It is defined within a block
- It is the block in which it is defined.
- It cannot be accessed from outside the block of its declaration.

- A block of code begins and ends with curly braces { }.
- It exists only while the block of code in which they are declared is executing.

2. Function scope: -

- It is declared within a function is extended to function block, and all sub-blocks.
- It is accessible in all the sub-blocks.
- The lifetime of a function scope variable is the lifetime of the function block.
- The scope of formal parameters is block function scope.

3. File scope: -

- A variable declared above all blocks and functions have the scope of a file.
- The file scope variable is the entire program.
- The lifetime of a file scope variable is the lifetime of a program.

4. Class scope: -

- A Class is a way to bind the data and its associated functions together.
- Classes provide a method for packing together.
-

2. Explain call by value method with an example? **[Oct-07,09,June-08], [March-2010]**

- In this method, the called function creates new variable to store the value of the arguments passed to it.
- In this method copies the value of actual parameters into the formal parameters.
- The function creates its own copy of arguments and then uses them.
- In this method, the flow of data is always from the call statement to the function definition.

Example: -

```
#include<iostream.h>
#include<conio.h>
void sawp(int n1,int n2)
{
int temp;
temp=n1;
n1=n2;
n2=temp;
cout<<n1<<'\t'<<n2<<'\n';
}
main()
{
```

```
int m1=10,m2=20;
clrscr();
cout<<"Before swap"<<m1<<'\t'<<m2<<'\n';
cout<<"Calling swap\n";
swap(m1,m2);
cout<<"Back to main values are"<<m1<<'\t'<<m2<<'\t';
getch();  }
```

Output:-

```
Before swap 10     20
Calling swap
20      10
Back to main values are  10      20
```

3. Explain call by reference method with suitable example? **[March-2008]**

In this method, the called function arguments-formal parameters become alias to the actual parameters in the calling function.
The function is working with its own arguments.
It is actually working on the original data.
In this method, any change made in the formal parameter is reflected back in the actual parameter.

Example: -

```
#include<iostream.h>
#include<conio.h>
void sawp(int &n1,int &n2)
{
int temp;
temp=n1;
n1=n2;
n2=temp;
cout<<n1<<'\t'<<n2<<'\n';
}
main()
{
int m1=10,m2=20;
clrscr();
cout<<"Before swap"<<m1<<'\t'<<m2<<'\n';
cout<<"Calling swap\n";
swap(m1,m2);
cout<<"Back to main values are"<<m1<<'\t'<<m2<<'\t';
getch(); }
```

Output:-

```
Before swap 10     20
Calling swap
 20     10
Back to main values are  10      20
```

4. Explain Inline Functions with an example? **[March-2007,2008], [June-2008],[October-2008]**

When the functions are small, the compiler replaces the function call statement by its definition ie, its code during program execution. This feature is called as inline function.

- An inline looks like a normal function in the source file but inserts the function's code directly into the calling program.
- Inline functions execute faster but require more memory space.
- Reusability of code leading to saving of memory space and reduction in code size

Example: (To see Page.No.113)

5. STRUCTURED DATA TYPE-ARRAYS

TWO MARK QUESTIONS: -

1. What is array? What are the different types? **[June-07, Oct-07, 08, 09]**

An array is a collection of variables of the same type that are referenced by a common name. Arrays are two types: One dimensional, Multi dimensional.

2. Give the syntax for Single Dimensional Array?

Syntax: - Datatype Array-identifier [size];
Example: - int mark[5];

3. How will you declare two-dimensional array? **[June-08, Mar-09, 10]**

Syntax: - Datatype Array-identifier [row] [column];
Example: - int mark[5][5];

4. What is sorting?

One can rearrange the data in a given array either in ascending or descending order. This process is called Sorting.

5. How the strings are treated? Give example?

- Strings are called as literals, which are treated as single dimension of characters.
- The declaration of strings is same as numeric array.

Example: - char name [10];
char vowels[ ] = {'a', 'e', 'i', 'o', 'u'};

6. Give the syntax for gets( ) & getline?

gets (char *) **getline:-** cin.getline(char *, no.of characters, delimiter);

7. What are the two methods to display the contents of character array?

a) cout<<name – this is similar to any other variable
b) cout.write (pincode, 7);

8. Write a note on write ()?
- It is a member function of standard output stream.
- All member functions of a class should be accessed through an object / instance of class.

The two parameters required for write function are identifier string characters and number of characters to be displayed.

9. Give the syntax for strlen( ) & its uses? **[June-08]**

strlen(char *)
It returns the number of characters stored in the array.
Ex: - na="Chennai"
p=strlen(na);
The given string length is 7

10. Give the syntax for strcpy( ) & its uses? **[Mar-09]**

strcpy(char * ,)
Copies source string to target string.
Ex: - a="Chennai"
strcpy(b,a);

11. Give the syntax for strcmp( ) & its uses? **[June-07,09, Oct-07, Mar-10]**

strcmp(char1,char2)
It compares the two given strings. It returns 0 if strings are equal
Ex:- strcmp("Abc","Abc">0)

12. Write about any two string functions in C++? **[Mar-08]**

Ans:- (Refer Q.No.9,10,11)

13. What is array of strings? Give example? **[Oct-08]**
- An array of strings is a two-dimensional character array.
- The size of first index(rows) determines the number of strings
- The size of second index(column) determines maximum length of each string

Example:-

```
Char day[7][10] = {"Sunday", "Monday", "Tuesday",
                   "Wednesday","Thursday", "Friday", "Saturday" };
```

14. Explain the memory representation of 2-D arrays? **[March-2007]**
- A 2-D array is stored in sequential memory blocks.
- The elements are stored either row-wise manner or column-wise manner

6. CLASSES & OBJECTS

TWO MARK QUESTIONS: -

1. What is a Class? Give example? **[Oct-08]**

A class is a new way of creating and implementing a user defined data type. Classes provide a method for packing together.

(Or)

A Class is a way to bind the data and its associated functions together.

Example :-

```
Class student
{
    char name[30];
    int rno, m1,m2,m3,tot_marks;
};
```

2. What is the specifying of a class? (or) What are the two parts of class declaration? **[Oct-07, Mar-10]**

1. Class Declaration 2. Class Function Definitions

3. What are the three access specifiers?

Private, Pubic and Protected

4. Write the general form of class declaration? **[Oct-07]**

```
Class class-name
{
private :
        variable declaration
        function declaration
protected :
        variable declaration
        function declaration
public :
        variable declaration
        function declaration
};
```

5. Define Encapsulation? **[June-09]**

The binding of data and functions together into single entity is referred to as encapsulation.

6. What is meant by Data hiding? **[June-07]**

The members and functions declared under private are not accessible by members outside the class, this is referred to as data hiding.

7. What is Data Abstraction? **[June-08, Oct-09,Mar-09]**

Instruments allowing only selected access of components to objects and to members of other classes is called as Data Abstraction.

8. Difference between Data members and Member functions?

Data Members	Member Functions
1. It is the data variables that represent the features or properties of a class.	1. It is the functions that perform specific tasks in a class.
2. It is also called as attributes.	2. It is called as methods.

9. Write about Static Data Member? **[June-07]**

1. It is initialized to zero, only when the first object of its class is created. No other initialization is permitted
2. Only one copy of the member variable is created and is shared by all the other objects of its class type..
3. Its scope or visibility is within the class, but its life time is the life time of the program

10. Define friend functions?

Accessible by only its own members and certain special functions called as friend functions.

11. What is the use of a dot operator? (or) How are the class members accessed? **[June-08]**

The members of a class are accessed using the dot operator. The call statement to the function execute ( ) of the class.

Ex:- Stud.execute( );

Where stud→ Member function

.→ Dot operator

execute ( )→ Object Name

12. What is meant by methods in C++?

- The class data type can be further extended by defining its associated functions.
- These functions are also called methods, as they define the various operations that can be performed on the data.

13. What a short note on a member of a class?

- Class comprised of members. Members are further classified as data members and member functions.
- Data members are the data variables that represent the features or properties of a class

- Member functions are the functions that perform specific tasks in a class.
- Member functions are called methods and data members are also called attributes.

14. What are the different ways of creating objects? **[June-09]**
    - Once a class has been declared, variables of that type can be declared.
    - 'stud' is a variable of type student, student is a data type of clas.
    - In C++ the class variables are known as objects.
    - The declaration of an object is similar to that of a variable of any basic type.
    - Objects can also be created by placing their names immediately after the closing brace of the class declaration.

15. Write short notes on memory allocation of objects? **[Mar-08,10]**
    - No separate memory space is allocated for member function when the object is created.
    - Memory space required for the member variables are only allocated separately for each object.
    - Separate memory allocations for the objects are essential because the member variables will hold different data values for different objects.

16. Write a note on array of objects in C++? **[Mar-07, Oct-09]**

```
class product
{
int code, quantity;
float price;
public :
  Void assign_data();
  Void display();
}p[3];
void main()
{
p[0].assign_data();
p[0].display();
}
```

17. Explain the access specifiers of class? **[Mar-09]**

    The three access specifiers are private, public and protected
    - The members that have been declared as private, can be accessed only from within the class.
    - The members that have been declared as protected, can be accessed from within the class, and the members of the inherited classes.

- The members that have been declared as public, can be accessed from outside the class also.

18. Write the characteristics of member functions? **[Mar-07]**
- Members functions can access the private data of a class. A non member function cannot do so
- A member function can call another member function directly, without using the dot operator. This is called as nesting of member functions
- Member functions can be of static type
- The return type of a member function can be of object data type

FIVE MARK QUESTION: -

1. Give the general form of a class and explain with an example?
(Refer Page.No.153)

7. POLYMORPHISM

TWO MARK QUESTIONS: -

1. Define a Polymorphism?
The word polymorphism means many forms (Poly-many, morph-Shapes).

2. Define Overloading?
The term overloading means a name having two or more distinct meanings.

3. What is function overloading? **[Oct-08]**
The ability of the function to process the message or data in more than one form is called as function overloading.

4. What is operator overloading? **[Mar-09]**
The mechanism of giving special meaning to an operator is called as operator overloading.

5. List out the operators that cannot be overloaded? **[June-08]**
1. Membership operator 2. Scope resolution operator
3. Size of operator 4. Conditional operator

6. How are functions invoked in function overloading?
[Mar-07,10, June-07,Oct-09]
- Look for the exact match of a function prototype with that of a function call statement.
- In case an exact match is not available, it looks for the next nearest match.

FIVE MARK QUESTIONS: -

1. What is the advantage of operator overloading? (Or) List out the rules for overloading operators? **[Mar-07, June-08, Oct-09]**
   - Only existing operators can be overloaded. New operators cannot be created.
   - The overloaded operator must have at least one operand of user-defined type.
   - The basic definition of an operator cannot be replaced.
   - Overloaded operators behave in the same way as the basic operators in terms of their operands.
   - When binary operators are overloaded, the left hand object must be an object of the relevant class
   - Binary operators overloaded through a member function take one explicit argument.

2. Explain function overloading with rules? **[June-07, 09, Oct-07, Mar-08,09]**

Function overloading :- The ability of the function to process the message or data in more than one form is called as function overloading.

Rules for function overloading:-

- Each overloaded function must differ either by the number of its formal parameters.
- The return type of overloaded functions may or may not be the same data type.
- The default arguments of overloaded functions are not considered by the C++ compiler as part of the parameter list.
- Do not use the same function name for two unrelated functions.

3. Explain function overloading with an example? **[Mar-2010]**

Function overloading: - The ability of the function to process the message or data in more than one form is called as function overloading.

Example:-

```
area_circle( )    // to calculate the area of a circle
area_triangle( )   // to calculate the area of a triangle
area_rectangle( ) // to calculate the area of a rectangle
```

The above three different prototype to compute area, for different shapes can be rewritten using a single function header.

```
float area(float radius);
float area(float half, float base, float height);
float area(float length, float breadth);
// Function Overloading
#include<iostream.h>
#include<conio.h>
float area(float r)
{  return(3.14*r*r);  }
```

```
float area(float hf, float b, float ht)
{   return(hf*b*ht);  }
float area(float le, float br)
{   return(le*br);     }
main()
{
float r,b,h;
int ch=0;
clrscr();
do
{
clrscr();
cout<<"1.Circle\n";
cout<<"2.Triangle\n";
cout<<"3.Rectangle\n";
cout<<"4.Exit\n";
cout<<"Enter your choice\n";
cin>>ch;
switch(ch)
{
case 1:
        cout<<"Enter Radius\n";
        cin>>r;
        cout<<"Area of Circle="<<area(r);
        getch();
        break;
case 2:
        cout<<"Enter Base, Height\n";
        cin>>b>>h;
        cout<<"Area of Triangle="<<area(0.5,b,h);
        getch();
        break;
case 3:
        cout<<"Enter Length, Breadth\n";
        cin>>h>>b;
        cout<<"Area of Rectangle="<<area(h,b);
        getch();
        break;
  }
 }while(ch<=3);
}
```

The first prototype had one argument, second one 3 arguments and the third one had 2 arguments.

8. CONSTRUCTORS & DESTRUCTORS

TWO MARK QUESTIONS: -

1. Difference between Constructors & Destructors? **[Mar-09]**

Constructors: -

- When an instance of a class comes into a scope, a special function called the constructors gets executed.
- It initializes the class object.

Destructors: -

- When a class object goes out of a scope, a special function called the destructor gets executed.
- Both the functions return nothing.

2. What are the functions of a constructor? **[Oct-07, June-08]**
   - The constructor function initializes the class object
   - The memory space is allocated to an object.

3. When is a Copy constructor executed? **[June-07, Oct-09]**
   - When an object is passed as a parameter to any of the member functions. **Ex:-** void add::putdata(add x);
   - When a member function returns an object. **Ex: -** add getdata();
   - When as object is passed by reference to constructor
     Ex: - add a;b(a);

4. Define Destructors?
   - It is a function that removes the memory of an object, which was allocated by the constructor at the time of creating an object.
   - It carries the same name as the class tag, but with a tilde (~) as prefix.

5. What is default contractor?
   The constructors add () is a constructor without parameters. It is called as default constructor.

6. What is Constructor?
   - The constructor is a special function that initializes objects when they are created.
   - It is automatically invoked when an object is created.

FIVE MARK QUESTIONS: -

1. What is a constructor? Explain the rule for Constructors? **[Mar-07, June-09]**

Constructor:-

- The constructor is a special function that initializes objects when they are created.
- It is automatically invoked when an object is created.

Functions of Constructor:-

- The constructor function initializes the class object
- The memory space is allocated to an object.

Rules for Constructor:-

- The name of the constructor must be same as that of the class
- A constructor can have parameter list
- The constructor function can be overloaded.
- The compiler generates a constructor, in the absence of a user defined constructor.
- The constructor is executed automatically

2. List the rules for Destructor? **[Mar-10]**
   - The destructor has the same name as that of the class prefixed by the tilde character (~).
   - The destructor cannot have arguments
   - It has no return type
   - Destructors cannot be overloaded
   - In the absence of user-defined destructor, it is generated by the compiler.
   - The destructor is executed automatically when the control reaches the end of class scope.

9. INHERITANCE

TWO MARK QUESTIONS: -

1. Define inheritance?
   - Inheritance is the most powerful feature of an object oriented programming language.
   - It is a process of creating new classes called derived classes, from the existing or base classes.

2. What are the advantages of inheritance? **[Oct-07, Mar-09]**

- **Reusability of code: -** Many applications are developed in an organization. Code developed for one application can be reused in another application. This saves a lot of development time.
- **Code sharing: -** The method of the base class can be shared by the derived class.
- **Consistency of interface: - The** inherited attributes and methods provide a similar interface to the calling methods.

2. Define a base class?
   - It is a class from which other classes are derived.
   - A derived class can inherit members of a base class.

3. What are the points should be observed while defining a derived class?
   - The keyword class has to be used.

- The derived class is used after the keyword class.
- A single colon
- The type of derivation private, public or protected
- The name of the base class or parent class
- The remainder of the derived class definition

4. Write the syntax for creating a derived class from base class? How the derived class should be indicated? **[Mar-08, Oct-09]**

```
Class der_name: visibility mode base class-id
    {
        data members of the derived_class
        functions members of derived_class
    }
```

5. What is accessibility?

An important feature in inheritance is to know as to when a number of a base class can be used by the objects or the members of derived class. This is called as accessibility.

6. What is an abstract class? **[June-08]**

Classes used only for deriving other classes are called abstract classes. ie., to say that objects for these classes are not declared.

7. What are the different types of inheritance? **[Mar-07, 10, June-07]**

Single inheritance, multiple inheritance, Multilevel inheritance, Hybrid inheritance and Hierarchical inheritance.

8. What is single inheritance?

When a sub class inherits only from the one base class, it is known as single inheritance. Ex: - Base Class→Employee Derived Class→Manager

9. What are the three access specifiers used to inherit a derived class?**[Oct-08]**
   - The three access specifiers are private, protected and public.
   - Access specifier is also referred to as visibility mode.
   - The default visibility mode is private.

10. How constructors and destructors are executed in inheritance?
   - The constructors are executed in the order of inherited class ie., from base constructor to derived.
   - The destructors are executed in the reverse order.

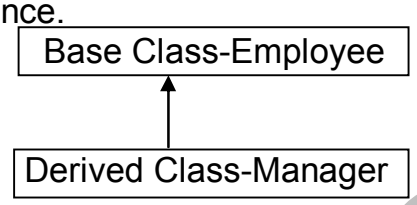
FIVE MARK QUESTION: -

1. Explain the different types of inheritance? **[Oct-08]**

There are different types of inheritance.
Single inheritance, Multiple inheritance, Multilevel inheritance, hybrid inheritance and Hierarchical inheritance

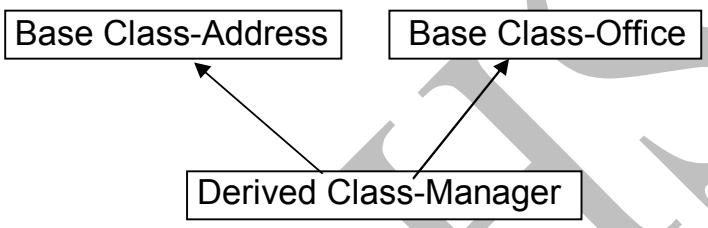
1.Single inheritance:-

When a derived class inherits from only one base class, it is called as single inheritance.



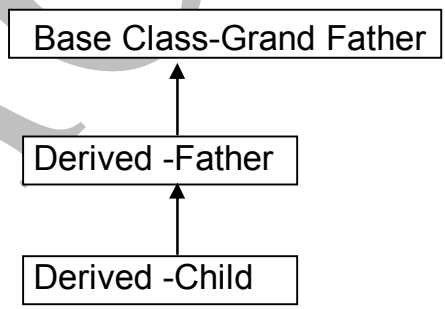
2.Multiple inheritance:-

When a derived class inherits from multiple base classes, it is called as single inheritance.



3.Multilevel inheritance:-

The transitive nature of inheritance is reflected by this from of inheritance. When a class is derived from a class, which is a derived class itself. It is called as multilevel inheritance.



10. IMPACT OF COMPUTERS ON SOCIETY

TWO MARK QUESTIONS: -

1. What is E-banking mean? **[Mar-09]**
   - E-banking permits banking from the comfort of the home by using internet facilities.
   - It has truly improved the reach and services of banks.

2. What is meant by e-Learning ? **[Mar-07]**

- e-Learning that enables online educational programs leading to degrees and certifications.

3. What is ATM? **[Mar-08]**
   - It means Automatic Teller Machine
   - It enables withdrawal of money from the accounts in a particular bank anytime and anywhere.
   - It helps the user in emergency situations where money is needed during the nights and holidays.

4. What is e-shopping?
   - You can purchase any product, any brand, any quantity from anywhere through e-shopping. You need not go to the shop.
   - The pictures and other details are available on the website of the shop.
   - You have to select and order.
   - Credit cards and prior registration with the shop are the popular methods.
   - The items purchased will be delivered at your home.

11. IT ENABLED SERVICES

TWO MARK QUESTIONS: -

1. What is meant by ITES? **[Mar-09]**
   - Information Technology that helps in improving the quality of service to the users is called IT Enabled Services.
   - ITES are human intensive services that are delivered over telecommunication networks.

2. What is a Call centers? **[June-07, Oct-09, Mar-10]**
   - A call center has adequate telecom facilities, trained consultants, access to wide database, Internet and other on-line information support to provide information and support services to customers.
   - It operates to provide round the clock and year round service ie, 24 x 365 service

3. What is Medical Transcriptions? **[June-08]**
   - It is a permanent, legal document that formally states the result of a medical investigation.
   - It facilities communication and supports the insurance claims.
   - There are three main steps in Medical Transcription.

4. What is meant by data digitization? **[Oct-08, June-09]**
   - It refers to the conversion of non-digital material to digital form.

- A wide variety of materials as diverse as maps, manuscripts, moving images and sound may be digitized.

5. Write the key benefits of data digitization? **[June-08]**
   - Long-term preservation of the documents.
   - Storage of important documents at one place.
   - Easy to use and access to the information
   - Easy transfer of information in terms of images and text
   - Easy transfer of information through CD-ROM, Internet and other electronic media.

6. Define Data Management? **[Oct-07]**
   - Data Management is a category of ITES pertaining to collection, digitization and processing of data coming from various sources.

12. COMPUTER ETHICS

TWO MARK QUESTIONS: -

1. What is meant by Computer Crime? **[June-09]**
   A computer crime is any illegal activity using computer software, data or access as the object, subject or instrument of the crime.

2. List out the common computer crimes? **[Mar-07]**
   1) Stealing hardware 2) Virus 3) Cracking
   4) Theft of computer time 5) Hardware and software piracy
   6) illegal access to confidential files

3. What is meant by Piracy? **[June-07]**
   Making and using duplicate hardware and software is called piracy.

4. Write a short note on Virus? **[Oct-07, Mar-10]**
   - A virus is a self-replicating program that can cause damage to data and files stored on your computer.
   - These are programs written by programmers with great programming skills are motivated by the need for a challenge.

5. What is cracking? **[Mar-08, Oct-08,09]**
   - It is the illegal access to the network or computer system.
   - Illegal use of special resources in the system is the key reason for cracking.
   - The resources may be hardware, software, files or system information.
   - Revenge, business reasons and thrill are other common reasons for committing this crime.

BLUE PRINT				
XII STANDARD COMPUTER SCIENCE				
CHAPTER	**VOL-I QUESTIONS**			
	1 Marks	**2 Marks**	**5 Marks**	**Total**
Chapter 1-5	9	2	2	23
Chapter 6	7	2	2	21
Chapter 7	9	2	1	18
Chapter 8	5	2	--	9
Chapter 9	5	2	--	9
Total	**35**	**20**	**25**	**80**
CHAPTER	**VOL-II QUESTIONS**			
	1 Marks	**2 Marks**	**5 Marks**	**Total**
Chapter 1	2	1	--	4
Chapter 2	4	3	--	10
Chapter 3	5	1	1	12
Chapter 4	4	1	1	11
Chapter 5	3	2	--	7
Chapter 6	6	2	--	10
Chapter 7	3	1	1	10
Chapter 8	4	1	1	11
Chapter 9	3	1	1	10
Chapter10-12	6	2	--	10
Total	**40**	**30**	**25**	**95**

Note :- Vol-II 2Marks

Unit 1 To 3 → 5 Questions
Unit 4 & 5 → 3 Questions
Unit 6 To 9 → 5 Questions

Vol- II Totally **15 Questions**

Unit 10 To 12→ 2 Questions

Note :- Vol-II 5Marks

Unit 3 → 1 Question
Unit 4 → 1 Question
Unit 7 → 1 Question
Unit 6,7,8,9→ 2 Questions
(1Debug & 1Output)

Vol- II Totally **5 Questions**