CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

CHAPTER – 13

INTRODUCTION TO OBJECT ORIENTED PROGRAMMING TECHNIQUES

SECTION – A

I. CHOOSE THE CORRECT ANSWER:

1. The term is used to describe a programming approach based on classes and objects is
   a) OOP b) POP c) ADT d) SOP
2. The paradigm which aims more at procedures is _____.
   a) Object Oriented Programming **b) Procedural programming**
   c) Modular programming d) Structural programming
3. Which of the following is a user defined data type?
   a) Class b) Float c) Int d) Object
4. The identifiable entity with some characteristics and behavior is.
   a) Class **b) Object** c) Structure d) Member
5. The mechanism by which the data and functions are bound together into a single unit is known as
   a) Inheritance **b) Encapsulation** c) Polymorphism d) Abstraction
6. Insulation of the data from direct access by the program is called as
   a) Data hiding b) Encapsulation c) Polymorphism d) Abstraction
7. Which of the following concept encapsulate all the essential properties of the object that are to be created?
   a) Class **b) Encapsulation** c) Polymorphism d) Abstraction
8. Which of the following is the most important advantage of inheritance?
   a) Data hiding **b) Code reusability** c) Code modification d) Accessibility
9. "Write once and use it multiple time" can be achieved by
   a) Redundancy **b) Reusability** c) Modification d) Composition
10. Which of the following supports the transitive nature of data?
    a) Inheritance b) Encapsulation c) Polymorphism d) Abstraction

CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

SECTION – B

II. ANSWER TO ALL THE QUESTIONS (2 MARKS):

1. How is modular programming different from procedural programming paradigm?

PROCEDURAL PROGRAMMING	MODULAR PROGRAMMING
⊕ Procedural means a list of instructions were given to the computer to do something.	⊕ Modular programming consists of a list of instructions that instructs the computer to do something.
⊕ Procedural programming aims more at procedures. This emphasis on doing things.	⊕ **Paradigm consists of multiple modules, each module has a set of functions of related types. Data is hidden under the modules.** Arrangement of data can be changed only by modifying the module

2. Differentiate classes and objects.

CLASSES	OBJECTS
⊕ Class is mechanism of binding data members and associated methods in a single unit. It belongs to data type.	⊕ Object is the instance of the class.
⊕ It generates the Objects.	⊕ It gives life to the class.
⊕ Memory space is not allocated, when it is created.	⊕ Memory space is allocated, when it is created.
⊕ It represents a group of similar objects	⊕ It is also called Class Variables.
⊕ **Eg:** ⊕ Car is the class.	⊕ **Eg:** ⊕ BMW, Audi are the objects.

3. What is Polymorphism?

⊕ Polymorphism is the ability of a message or function to be displayed in more than one form.

4. How Encapsulation and Abstraction is are interrelated?

⊕ Abstraction and Encapsulation look **similar** to each other by its definition.

- ⊕ The standard definition of abstraction is showing **only necessary features** to the user and hides its implementation. On the other hand, Encapsulation is hiding the complexity from the outer world.
- ⊕ **Encapsulation** is **wrapping data** into **single** unit (e.g A class). **Abstraction** is **hiding unessential parts** and showing only essential data.

5. Write the disadvantages of OOP.

- ⊕ **Size:** Object Oriented Programs are much larger than other programs.
- ⊕ **Effort:** Object Oriented Programs require a lot of work to create.
- ⊕ **Speed:** Object Oriented Programs are slower than other programs, because of their size.

SECTION – C

III. ANSWER TO ALL THE QUESTIONS (3 MARKS):

1. What is paradigm? Mention the different types of paradigm.

- ⊕ **Paradigm means organizing principle of a program. It is an approach to programming.**
- ⊕ There are different approaches available for problem solving using computer.
- ⊕ **Procedural programming:** It aims more at procedures.
- ⊕ **Modular Programming:** It consists of multiple modules. Each module has a set of functions of related types
- ⊕ **Object Oriented Programming:** It is widely accepted that object-oriented programming is the most important and powerful way of creating software.

2. Write a note on the features of procedural programming.

- ⊕ Programs are organized in the form of **subroutines** or **sub programs**
- ⊕ All data items are **global**
- ⊕ Suitable for **small** sized software application
- ⊕ Difficult to maintain and enhance the program code as any change in data type needs to be propagated to all subroutines that use the same data type. This is time consuming.
- ⊕ Example: **FORTRAN** and **COBOL**.

3. List some of the features of modular programming

- ⊕ Emphasis on algorithm rather than data
- ⊕ Programs are divided into **individual modules**
- ⊕ Each module is independent of each other and have their **own local data**
- ⊕ **Modules** can work with its own data as well as with the data passed to it.
- ⊕ Example: **Pascal** and **C**

4. What do you mean by modularization and software reuse?

- ⊕ **Modularization:** Where the program can be decomposed into **modules**.
- ⊕ **Software re-use:** Where a program can be composed from existing and new modules.

5. Define information hiding.

- ⊕ Encapsulation is the most striking feature of a class.
- ⊕ The data is not accessible to the outside world, and only those functions which are wrapped in the class can access it.
- ⊕ These functions provide the interface between the object's data and the program.
- ⊕ **This encapsulation of data from direct access by the program is called data hiding or information hiding.**

SECTION – D

IV. ANSWER TO ALL THE QUESTIONS (5 MARKS):

1. Write the differences between Object Oriented Programming and procedural programming

	OBJECT ORIENTED PROGRAMMING	PROCEDURAL PROGRAMMING
Expansion	OOP is Object Oriented Programming	POP is Procedural Oriented Programming
Approach	It uses Bottom Up Approach.	It uses Top Down Approach.
Division	In OOP, the main modules in a program are classes, rather than procedures.	In POP, modules are procedures, where a procedure is a sequence of statements.
Importance	In OOP, Importance is given to the data rather than procedures or functions because it works as a **real world**.	In POP, Importance is not given to **data** but to functions as well as **sequence** of actions to be done.
Access	In OOP, there are 3 Access Specifiers private,	In POP, there are no Access Specifier

Specifiers	protected and public.	
Overloading	In OOP, the Overloading is possible.	In POP, the Overloading is not possible.
Data Hiding	In OOP, the Data Hiding is possible.	In POP, the Data Hiding is not possible.
Addition	In OOP, the Addition of new data and function is easy.	In POP, the Addition of new data and function is not easy.
Definition	The object-oriented approach lets you create classes and objects that model real world objects.	Procedures are a sequence of imperative statements, such as assignments, tests, loops and invocations of sub procedures. These procedures are functions, which map arguments to return statements.
Examples	C++, Java, VB.NET	FORTRAN, C, Pascal

2. What are the advantages of OOPs? (ANY THREE)

Re-usability:

⊕ **"Write once and use it multiple times"** you can achieve this by using class.

Redundancy:

⊕ Inheritance is the good feature for data redundancy. If you need a same functionality in multiple classes you can write a common class for the **same functionality and inherit that class to sub class.**

Easy Maintenance:

⊕ It is easy to maintain and modify existing code as new objects can be created with small differences to existing ones.

Security:

⊕ Using **data hiding** and **abstraction** only necessary data will be provided thus maintains the security of data.

3. Write a note on the basic concepts that support OOPs?

ENCAPSULATION:

- **Encapsulation** is about binding the data variables and functions together in class. It can also be called **Data binding.**
- Encapsulation is the most striking feature of a class.
- The data is not accessible to the outside world, and only those functions which are wrapped in the class can access it. These functions provide the interface between the object's data and the program.
- **This encapsulation of data from direct access by the program is called Data hiding or Information hiding.**

DATA ABSTRACTION

- **Abstraction** refers to showing only the essential features without revealing background details.
- **Classes** use the concept of abstraction to define a list of abstract attributes and function which operate on these attributes.
- They encapsulate all the essential properties of the object that are to be created.
- The attributes are called **Data members** because they hold information. The functions that operate on these data are called **Methods or Member function.**

MODULARITY

- **Modularity** is designing a system that is divided into a set of functional units (named modules) that can be composed into a larger application.

INHERITANCE

- **Inheritance** is the technique of building new classes **(derived class)** from an existing Class **(base class)**. The most important advantage of inheritance is **code reusability.**

POLYMORPHISM

- **Polymorphism** is the ability of a message or function to be displayed in more than one form.

CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

CHAPTER – 14

CLASSES AND ITS OBJECTS

SECTION – A

I. CHOOSE THE CORRECT ANSWER

1. The variables declared inside the class are known as data members and the functions are known as

   a) Data functions b) Inline functions **c) Member Functions** d) attributes

2. Which of the following statements about member functions are true or false?

   i) A member function can call another member function directly with using the dot operator.

   ii) Member function can access the private data of the class.

   a) i-True, ii-True b) i-False, ii-True c) i-True, ii-False d) i-False, ii-False

3. A member function can call another member function directly, without using the dot operator called as

   a) Sub function b) Sub member

   c) Nesting of member function d) Sibling of member function

4. The member function defined within the class behave like

   a) Inline functions b) Non inline function c) Outline function d) Data function

5. Which of the following Access Specifier protects data from inadvertent modifications?

   a) Private b) Protected c) Public d) Global

```
class x
{
int y;
public:
x(int z)
{
y=z;
}
} x1[4];
int main()
{
x x2[10];
return 0;
```

}

6. How many objects are created for the above program

a) 10 **b) 14** c) 5 d) 2

7. State whether the following statements about the constructor are true or false.

i) Constructors should be declared in the private section.

ii) Constructors are invoked automatically when the objects are created.

a) True, True b) True, False **c) False, True** d) False, False

8. Which of the following constructor is executed for the following prototype?

add display(add &); // add is a class name

a) Default constructor b) Parameterized constructor **c) Copy constructor** d) Non Parameterized constructor

9. What happens when a class with parameterized constructors and having no default constructor is used in a program and we create an object that needs a zero-argument constructor?

a) Compile-time error b) Domain error c) Runtime error d) Runtime exception.

10. Which of the following create a temporary instance?

a) Implicit call to the constructor b) Explicit call to the constructor

c) Implicit call to the destructor **d) Explicit call to the destructor**

SECTION – B

II. ANSWER TO ALL THE QUESTIONS (2 MARKS):

1. What are called members?

- ⊕ Class comprises of **members**.
- ⊕ Members are classified as **Data Members** and **Member functions**.
- ⊕ Data members are the data variables that represent the **features or properties of a class**. Data members are also called as **attributes.**
- ⊕ Member functions are the functions that perform **specific tasks in a class.** Member functions are called as **methods.**

2. Differentiate structure and class though both are user defined data type. (Any Four)

STRUCTURE	CLASS
⊕ Structure has all members by default **Public**	⊕ Class has all members by default **Private**

CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

⊕ Structure **cannot** be **inherited**	⊕ Class can be **inherited**
⊕ Structure **does not support data hiding**	⊕ Class **supports data hiding**
⊕ Structure belongs to **Value** type	⊕ Class belongs to **Reference** type
⊕ Structure contains only **Data Member**	⊕ Class contains both **Data Member and Member Functions**
⊕ Structure are stored on **Stack** on Memory	⊕ Class are stored on **Heap** on Memory

3. What is the difference between the class and object in terms of OOP?

CLASS	OBJECT
⊕ Class is mechanism of binding data members and associated methods in a single unit. It belongs to data type.	⊕ Object is the instance of the class.
⊕ It generates the Objects.	⊕ It gives life to the class.
⊕ Memory space is not allocated, when it is created.	⊕ Memory space is allocated, when it is created.
⊕ It represents a group of similar objects	⊕ It is also called Class Variables.
⊕ **Eg:** ⊕ Car is the class.	⊕ **Eg:** ⊕ BMW, Audi are the objects.

4. Why it is considered as a good practice to define a constructor though compiler can automatically generate a constructor?

- ⊕ To allocate the memory space for an object.
- ⊕ To initialize the data member of the class object.

5. Write down the importance of destructor.

- ⊕ The purpose of the destructor is to free the resources that the object may have acquired during its lifetime.
- ⊕ A **destructor** function **removes** the **memory** of an **object** which was **allocated** by the **constructor** at the time of creating an object.

SECTION – C

CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

III. ANSWER TO ALL THE QUESTIONS (3 MARKS):

1. Rewrite the following program after removing the syntax errors if any and underline the errors:

```
#include<iostream>
#include<stdio.h>
class mystud
{
int studid =1001;
char name[20];
public
mystud( )
{
}
void register ( )
{
cin>>stdid;gets(name);
}
void display ( )
{
cout<<studid<<": "<<name<<endl;
}
}
int main( )
{
mystud MS;
register.MS( );
MS.display( );
}
```

ANSWER:

```
#include<iostream> //Hash is missing
#include<stdio.h>
class mystud
{
int studid =1001;
char name[20];
public:  //Colon is missing
mystud( )
{
}
void register ( )
{
cin>>studid; //data member name is wrong studid
gets(name);
}
void display ( )
{
cout<<studid<<": "<<name<<endl;
}
}; //class is not terminated
int main( )
{
mystud MS;
MS.register(); //Object can call the member function using dot operator
MS.display( );
}
```

2. **Write with example how will you dynamically initialize objects?**

   ⊕ When the initial values are provided during runtime then it is called **Dynamic Initialization.**

PROGRAM FOR DYNAMIC INITIALIZATION:

```
#include<iostream>
using namespace std;
class X
{
    int n;
    float avg;
public:
X(int p,float q)
    {
    n=p;
    avg=q;
    }
void disp()
    {
    cout<<"\n Roll numbe:- " <<n;
    cout<<"\nAverage :- "<<avg;
    }
};
int main()
{
    int a ;
    float b;
    cout<<"\nEnter the Roll Number";
    cin>>a;
    cout<<"\nEnter the Average";
    cin>>b;
    X x(a,b); // dynamic initialization
    x.disp();
    return 0;
}
```

OUTPUT:

```
Enter the Roll Number 1201
Enter the Average 98.6
Roll numbe:- 1201
Average :- 98.6
```

3. What are advantages of declaring constructors and destructor under public accessibility?

ADVANTAGES:

- ⊕ Generally, a constructor and destructor should be defined under the public section of a class, so that its objects can be created and destroyed in any function.
- ⊕ It means that the constructor and destructor can be called directly by code in main() functions.

4. Given the following C++ code, answer the questions (i) & (ii).

```
class TestMeOut
{
public:
~TestMeOut() //Function 1
{
cout<<"Leaving the examination hall"<<endl;
}
TestMeOut() //Function 2
{
cout<<"Appearing for examination"<<endl;
}
void MyWork() //Function 3
{
cout<<"Attempting Questions//<<endl;
}
};
```

ANSWER:

(i) In Object Oriented Programming, what is Function 1 referred as and when does it get invoked / called?

- ⊕ The Function 1 is referred as Destructor.
- ⊕ When the class object goes out of scope, the destructor gets executed.
- ⊕ The destructor is a special member function because it has the same name as the class name but it is prefixed with tilde symbol (~).
- ⊕ This function executes at last

- ⊕ It is automatically executed.

(ii) **In Object Oriented Programming, what is Function 2 referred as and when does it get invoked / called?**

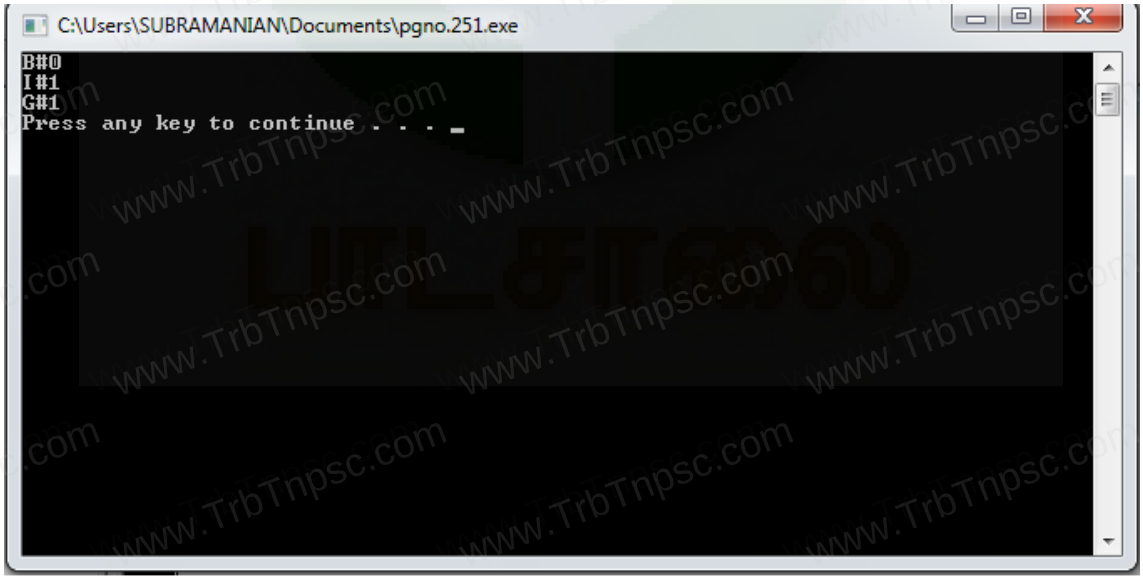
- ⊕ The Function 1 is referred as Constructor.
- ⊕ When the class object comes into the class, the constructor gets executed.
- ⊕ The constructor is a special member function because it has the same name as the class name.
- ⊕ This function executes first.
- ⊕ It is automatically executed.

5. Write the output of the following C++ program code:

```
#include<iostream>
using namespace std;
class Calci
{
char Grade;
int Bonus;
public:
Calci()
{
Grade='E';
Bonus=0;
} //ascii value of A=65
void Down(int G)
{
Grade-=G;
}
void Up(int G)
{
Grade+=G;
Bonus++;
}
```

```
void Show()
{
cout<<Grade<<"#"<<Bonus<<endl;
}
};
int main()
{
Calci c;
c.Down(3);
c.Show();
c.Up(7);
c.Show();
c.Down(2);
c.Show();
return 0;
}
```

OUTPUT:



CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

SECTION – D

IV. ANSWER TO ALL THE QUESTIONS (5 MARKS):

1. Explain nested class with example.

⊕ When one class becomes the member of another class then it is called Nested class and the relationship is called containership.

CLASSES CAN BE NESTED IN TWO WAYS.

⊕ By defining a class within another class

⊕ By declaring an object of a class as a member to another class

DEFINING A CLASS WITH IN ANOTHER

⊕ When a class is declared with in another class, the inner class is called as Nested class (ie the inner class) and the outer class is known as Enclosing class.

⊕ Nested class can be defined in private as well as in the public section of the Enclosing class.

PROGRAM:

```
#include<iostream>
using namespace std;
class enclose
{
private:
int x;
class nest
{
private :
int y;
public:
int z;
void prn()
{
y=3;z=2;
cout<<"\n The product of"<<y<<'*'<<z<<"= "<<y*z<<"\n";
}
}; //inner class definition over
```

```
nest n1;
public:
nest n2;
void square()
{
n2.prn(); //inner class member function is called by its object
x=2;
n2.z=4;
cout<<"\n The product of " <<n2.z<<'*'<<n2.z<<"= "<<n2.z*n2.z<<"\n";
cout<<"\n The product of " <<x<<'*'<<x<<"= "<<x*x;
}
}; //outer class definition over
int main()
{
enclose e;
e.square(); //outer class member function is called
}
```

OUTPUT:

```
The product of 3*2=6
The product of 4*4=16
The product of 2*2=4
```

BY DECLARING AN OBJECT OF A CLASS AS A MEMBER TO ANOTHER CLASS

⊕ Whenever an object of a class is declared as a member of another class it is known as a container class. In the container-ship the object of one class is declared in another class.

PROGRAM:

```
#include<iostream>
using namespace std;
class outer
{
```

```
int data;
public:
void get();
};
class inner
{
int value;
outer ot; // object ot of class outer is declared in class inner
public:
void getdata();
};
void outer :: get()
{
cout<<"\nEnter a value";
cin>>data;
cout<<"\nThe given value is "<<data;
}
void inner :: getdata()
{
cout<<"\nEnter a value";
cin>>value;
cout<<"\nThe given value is "<<value;
ot.get(); //calling of get() of class outer in getdata() of class inner
}
int main()
{
inner in;
in.getdata();
return 0;
}
```

OUTPUT:

Enter a value10

The given value is 10

Enter a value 20

The given value is 20

2. Mention the differences between constructor and destructor

CONSTRUCTOR	DESTRUCTOR
⊕ It can have arguments	⊕ It cannot have arguments
⊕ It can be overloaded	⊕ It cannot be overloaded
⊕ The name of the constructor must be same as that of the class name	⊕ The name of the destructor must be same as that of the class name but prefixed with Tilde Character (~)
⊕ There can more than one constructor in a class	⊕ There should be only one destructor in a class
⊕ It has no return type	⊕ It has no return type
⊕ It is executed automatically when the object is created.	⊕ It is executed automatically when the control reaches the end of the class.
⊕ In the absence of User – Defined constructor the compiler generates the constructor	⊕ In the absence of User – Defined destructor the compiler generates the constructor

3. Define a class RESORT with the following description in C++:

Private members:

Rno // Data member to store room number

Name //Data member to store user name

Charges //Data member to store per day charge

Days //Data member to store the number of days

Compute ( ) // A function to calculate total amount as Days * Charges and if the

//total amount exceeds 11000 then total amount is 1.02 * Days *Charges

Public member:

getinfo( ) // Function to Read the information like name , room no, charges and days

dispinfo ( ) // Function to display all entered details and total amount calculated

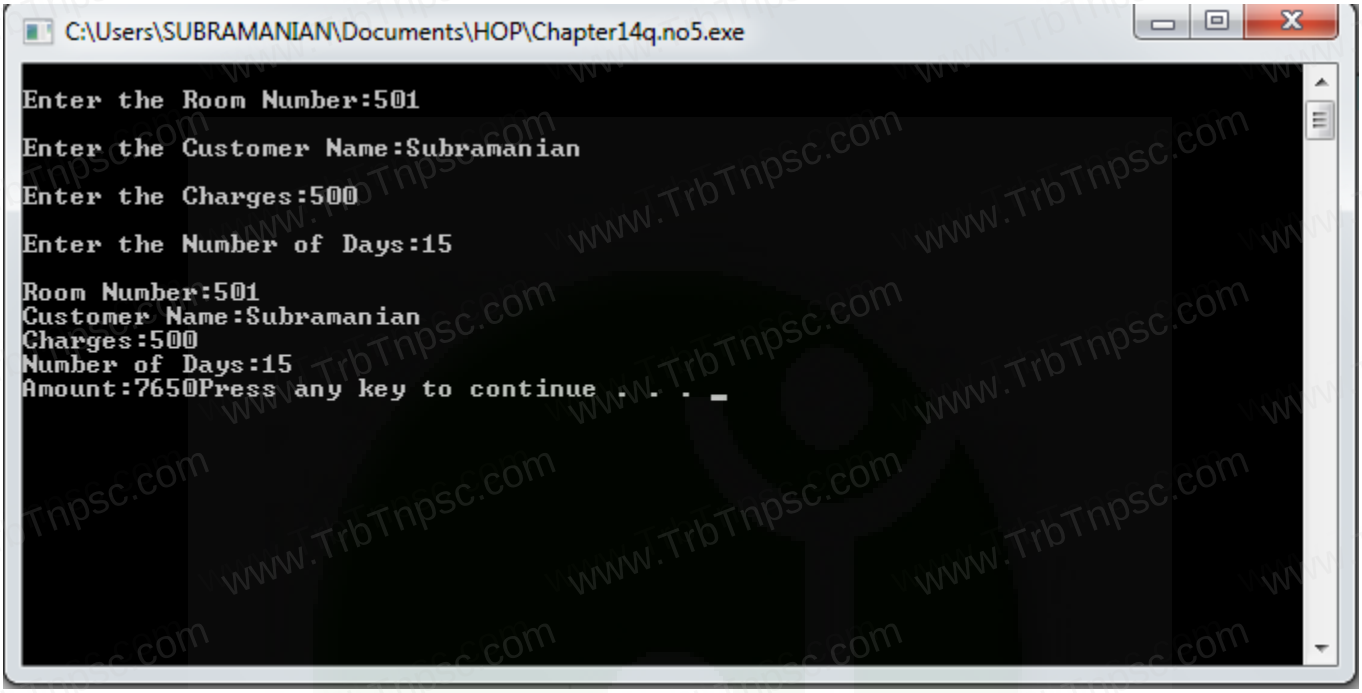
//using COMPUTE function

PROGRAM:

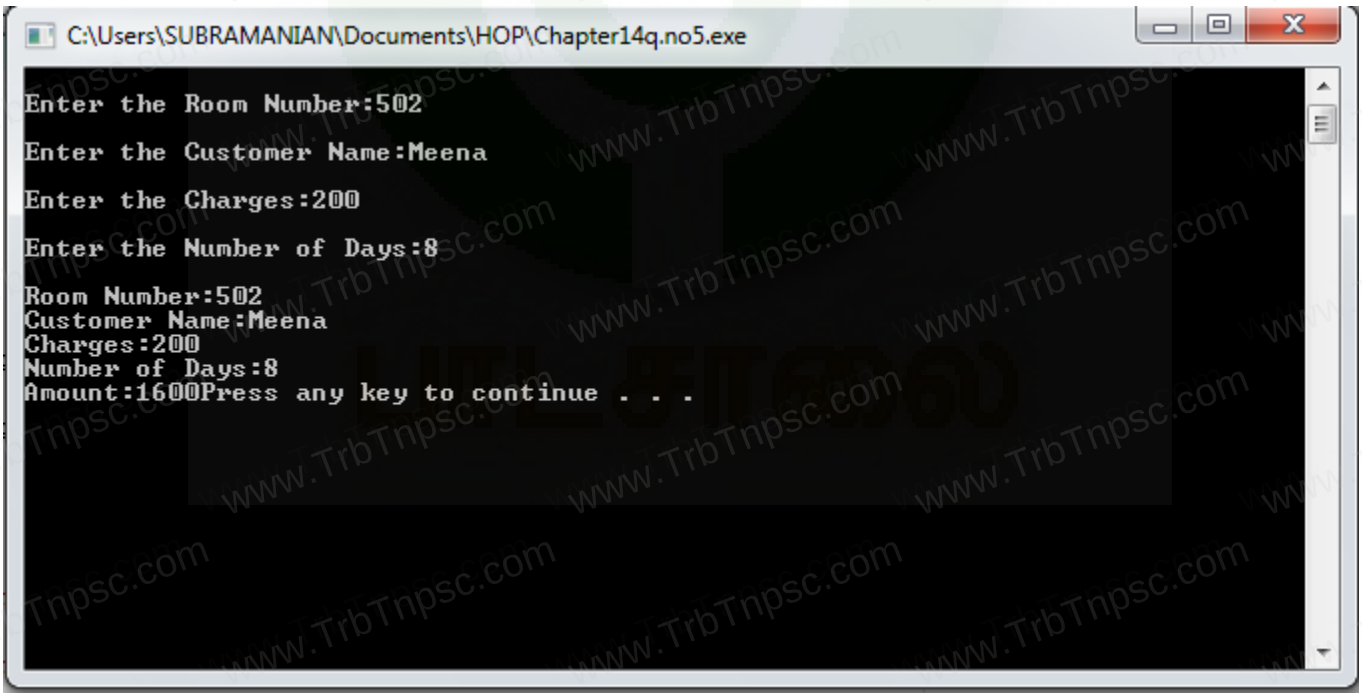
```
#include<iostream>
using namespace std;
class resort
{
int rno,days;
char rname[25];
float charges,amount;
void compute()
{
    amount=days*charges;
    if(amount>=5000)
    amount=1.02*charges*days;
}
public:
     void getinfo()
     {
         cout<<"\nEnter the Room Number:";
         cin>>rno;
         cout<<"\nEnter the Customer Name:";
         cin>>rname;
         cout<<"\nEnter the Charges:";
         cin>>charges;
         cout<<"\nEnter the Number of Days:";
         cin>>days;
         compute();
     }
     void displayinfo()
     {
         cout<<"\nRoom Number:"<<rno;
         cout<<"\nCustomer Name:"<<rname;
         cout<<"\nCharges:"<<charges;
         cout<<"\nNumber of Days:"<<days;
         cout<<"\nAmount:"<<amount;
     }
};
int main()
{
         resort r;
         r.getinfo();
         r.displayinfo();
         system("pause");
         return 0;
```

```
}
```

OUTPUT – 1:



OUTPUT – 2:



4. Write the output of the following

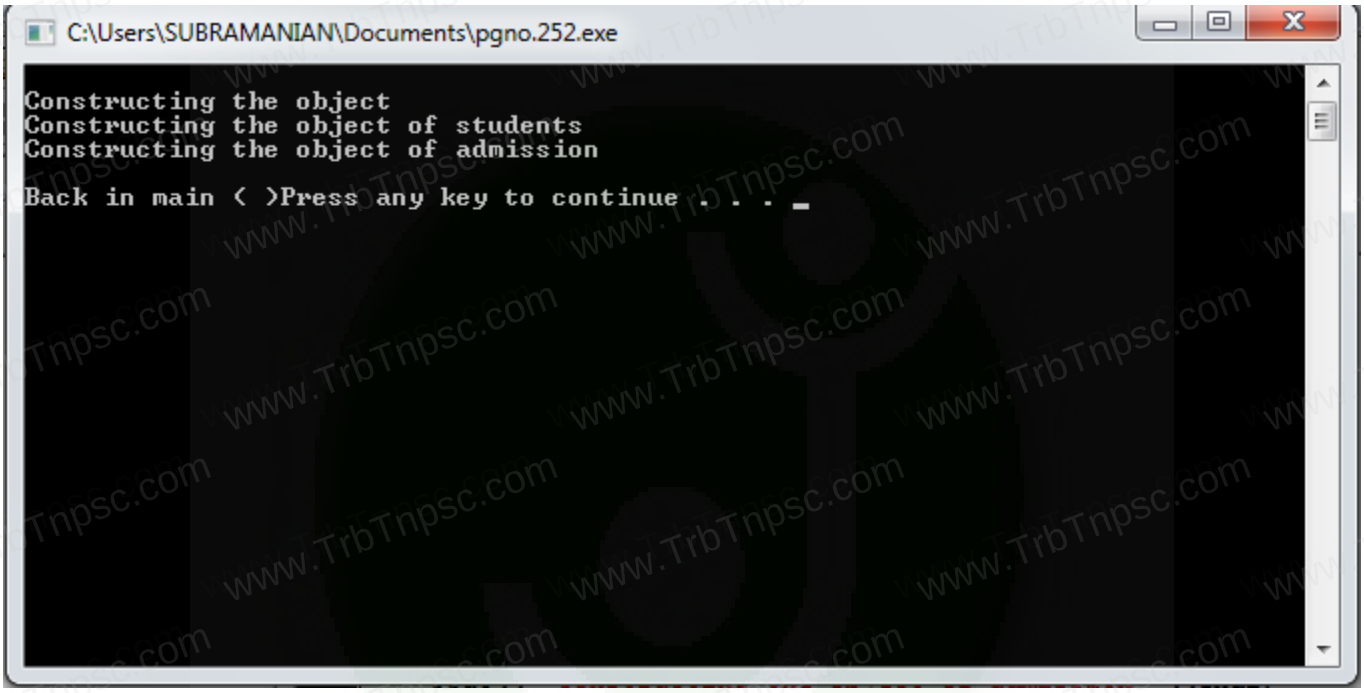
```
#include<iostream>
```

```
#include<stdio.h>
using namespace std;
class sub
{
int day, subno;
public :
sub(int,int); // prototype
void printsub()
{
cout<<" subject number : "<<subno;
cout<<" Days : " <<day;
}
};
sub::sub(int d=150,int sn=12)
{
cout<<endl<<"Constructing the object "<<endl;
day=d;
sub no=sn;
}
class stud
{
int rno;
float marks;
public:
stud( )
{
cout<<"Constructing the object of students "<<endl;
rno=0;
marks=0.0;
}
void getval()
```

```
{
cout<<"Enter the roll number and the marks secured ";
cin>>rno>>marks;
}
void printdet()
{
cout<<"Roll no : "<<rno<<"Marks : "<<marks<<endl;
}
};
class addmission
{
sub obj;
stud objone;
float fees;
public :
add mission ( )
{
cout<< "Constructing the object of admission "<<endl;
fees=0.0;
}
void printdet( )
{
objone.printdet();
obj.printsub( );
cout<<"fees : "<<fees<<endl ;
}
};
int main()
{
system("cls");
addmission adm;
```

```
cout<<endl<< "Back in main ( )";
return 0;
}
```

OUTPUT:



5. Write the output of the following

```
#include<iostream>
#include<stdio.h>
using namespace std;
class P
{
public:
P ( )
{
cout<< "\nConstructor of class P ";
}
~ P ( )
```

```
{
cout<< "\nDestructor of class P ";
}
};
class Q
{
public:
Q( )
{
cout<<"\nConstructor of class Q ";
}
~ Q( )
{
cout<< "\nDestructor of class Q ";
}
};
class R
{
P obj1, obj2;
Q obj3;
public:
R ( )
{
cout<< "\nConstructor of class R ";
}
~ R ( )
{
cout<< "\nDestructor of class R ";
}
};
int main ( )
```

```
{
Ro R;
Q oq;
P op;
return 0;
}
```

OUTPUT:

```
Constructor of class P
Constructor of class P
Constructor of class Q
Constructor of class R
Constructor of class Q
Constructor of class P
Destructor of class P
Destructor of class Q
Destructor of class R
Destructor of class Q
Destructor of class P
Destructor of class P
```

CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

CHAPTER 15

POLYMORPHISM

SECTION – A

I. CHOOSE THE CORRECT ANSWERS:

1. Which of the following refers to a function having more than one distinct meaning?

   a) Function overloading b) Member overloading

   c) Operator overloading d) Operations overloading

2. Which of the following reduces the number of comparisons in a program?

   a) Operator overloading b) Operations overloading

   c) Function overloading d) Member overloading

```
void dispchar(char ch='$',int size=10)
{
        for(int i=1;i<=size;i++)
        cout<<ch;
}
```

3. How will you invoke the function dispchar() for the following input?

   To print $ for 10 times

   a) dispchar(); b) dispchar(ch,size); c) dispchar($,10); d)dispchar('$',10 times);

4. Which of the following is not true with respect to function overloading?

   a) The overloaded functions must differ in their signature.

   b) **The return type is also considered for overloading a function.**

   c) The default arguments of overloaded functions are not considered for Overloading.

   d) Destructor function cannot be overloaded.

5. Which of the following is invalid prototype for function overloading

   a) void fun (intx); **b) void fun (intx);** c) void fun (double d); d) void fun (double d);

   void fun (char ch) ; **void fun (inty);** void fun (char ch); void fun (inty);

6. Which of the following function(s) combination cannot be considered as overloaded function(s) in the given snippet?

```
void print(char A,int B); // F1
void printprint(int A, float B); // F2
```

void Print(int P=10); // F 3

void print(); // F4

a) F1,F2,F3,F4 **b) F1,F2,F3** c) F1,F2,F4 d) F1,F3,F4

7. Which of the following operator is by default overloaded by the compiler?

a) * b) + c) += **d) = =**

8. Based on the following program answer the questions (8) to (10)

```
#include<iostream>
using namespace std;
class Point
{
        private:
        int x, y;
        public:
        Point(int x1,int y1)
        {
        x=x1;y=y1;
        }
        void operator+(Point &pt3);
        void show() {cout << "x = " << x << ", y = " << y; }
};
void Point::operator+(Point &pt3)
{
        x += pt3.x;
        y += pt3.y;
}
int main()
{
        Point pt1(3,2),pt2(5,4);
        pt1+pt2;
        pt1.show();
        return 0;
```

}

9. Which of the following operator is overloaded?

a) + b) operator c) :: d) =

10. Which of the following statement invoke operator overloading?

a) pt1+pt2; b) Point pt1(3,2),pt2(5,4); c) pt1.show(); d) return 0;

11. What is the output for the above program?

a) x=8, y=6 b) x=14, y=14 c) x=8, y=6 d) = x=5, y=9

SECTION – B

II. ANSWER TO ALL THE QUESTIONS (2 MARKS):

1. What is function overloading?

- ⊕ The ability of the function to process the message or data in more than one form is called as function overloading.
- ⊕ In other words function overloading means two or more functions in the same scope share the same name but their parameters are different.
- ⊕ The functions that share the same name are said to be overloaded and the process is called function overloading.

2. List the operators that cannot be overloaded.

- ⊕ Operator that are not overloaded are follows
  - ✓ scope operator **::**
  - ✓ sizeof
  - ✓ member selector **.**
  - ✓ member pointer selector *****
  - ✓ ternary operator **?:**

3. class add{int x; public: add(int)}; Write an outline definition for the constructor.

```
add::add(int y)
{
cout<<"Constructor for the Class Add";
y=x;
```

}

4. Does the return type of a function help in overloading a function?

⊕ No, the return type of a function does not help in overloading a function

5. What is the use of overloading a function?

⊕ Function overloading is not only implementing polymorphism but also reduces the number of comparisons in a program and makes the program to execute faster.

⊕ It also helps the programmer by reducing the number of function names to be remembered.

SECTION – C

III. ANSWER TO ALL THE QUESTIONS (3 MARKS):

1. What are the rules for function overloading?

⊕ The overloaded function must differ in the number of its arguments or data types

⊕ The return type of overloaded functions are not considered for overloading same data type

⊕ The default arguments of overloaded functions are not considered as part of the parameter list in function overloading.

2. How does a compiler decide as to which function should be invoked when there are many functions? Give an example.

⊕ The number and types of a function's parameters are called the **function's signature.**

⊕ When you call an overloaded function, the compiler determines the most appropriate definition to use, by comparing the argument types you have used to call the function with the parameter types specified in the definitions.

⊕ The process of selecting the most appropriate overloaded function or operator is called **overload resolution**

3. What is operator overloading? Give some example of operators which can be overloaded.

DEFINITION:

⊕ The mechanism of giving special meaning to an operator is known as operator overloading

- ⊕ The term operator overloading, refers to giving additional functionality to the normal C++ operators like +,++,-,—,+=,-=,*.<,>. It is also a type of polymorphism in which an operator is overloaded to give user defined meaning to it .
- ⊕ Eg: '+' operator can be overloaded to perform addition on various data types, like for Integer, String (concatenation) etc. Almost all operators can be overloaded in C++.

4. Discuss the benefit of constructor overloading?

- ⊕ It gives the flexibility of creating multiple types of objects of a class by having more number of constructors in a class, called **Constructor Overloading.**
- ⊕ In fact, it is similar to c++ function overloading that is also known as compiling time polymorphism.
- ⊕ Hence, if we want to construct an object in different way then we need constructor overloading in C++.

5. class sale { int cost, discount ;public: sale(sale &) }; Write a non inline definition for constructor specified;

sale::sale(sale &s)

```
{
    cost=s.cost;
    discount=s.discount;
 }
```

SECTION – D

IV. ANSWER TO ALL THE QUESTIONS (5 MARKS):

1. What are the rules for operator overloading?

- ⊕ Precedence and Associativity of an operator cannot be changed.
- ⊕ No new operators can be created, only existing operators can be overloaded.
- ⊕ One cannot redefine the meaning of an operator's procedure. You cannot change how integers are added. Only additional functions can be to an operator
- ⊕ Overloaded operators cannot have default arguments.
- ⊕ When binary operators are overloaded, the left hand object must be an object of the relevant class

2. Answer the question (i) to (v) after going through the following class.

```
class Book
{
intBookCode ; char Bookname[20];
float fees;
public:
Book( ) //Function 1
{
fees=1000;
BookCode=1;
strcpy (Bookname,"C++");
}
void display(float C) //Function 2
{
cout<<BookCode<<":"<<Bookname<<":"<<fees<<endl;
}
~Book( ) //Function 3
{
cout<<"End of Book Object"<<endl;
}
Book (int SC, char S[ ],float F) ; //Function 4
};
```

(i) In the above program, what are Function 1 and Function 4 combined together referred as?

- ⊕ Function 1 is called as Default Constructor
- ⊕ Function 4 is called as Parameterized Constructor

(ii) Which concept is illustrated by Function3? When is this function called/ invoked?

- ⊕ Function3 is called as Destructor.
- ⊕ When the class object goes out of scope this function is executed.

(iii) What is the use of Function3?

- The memory which was allocated for the object by the constructor function is removed by the Function3

(iv) Write the statements in main to invoke function1 and function2

- Book b1
- B1.display(999.99);

(v) Write the definition for Function4.

- **Function Definition:**

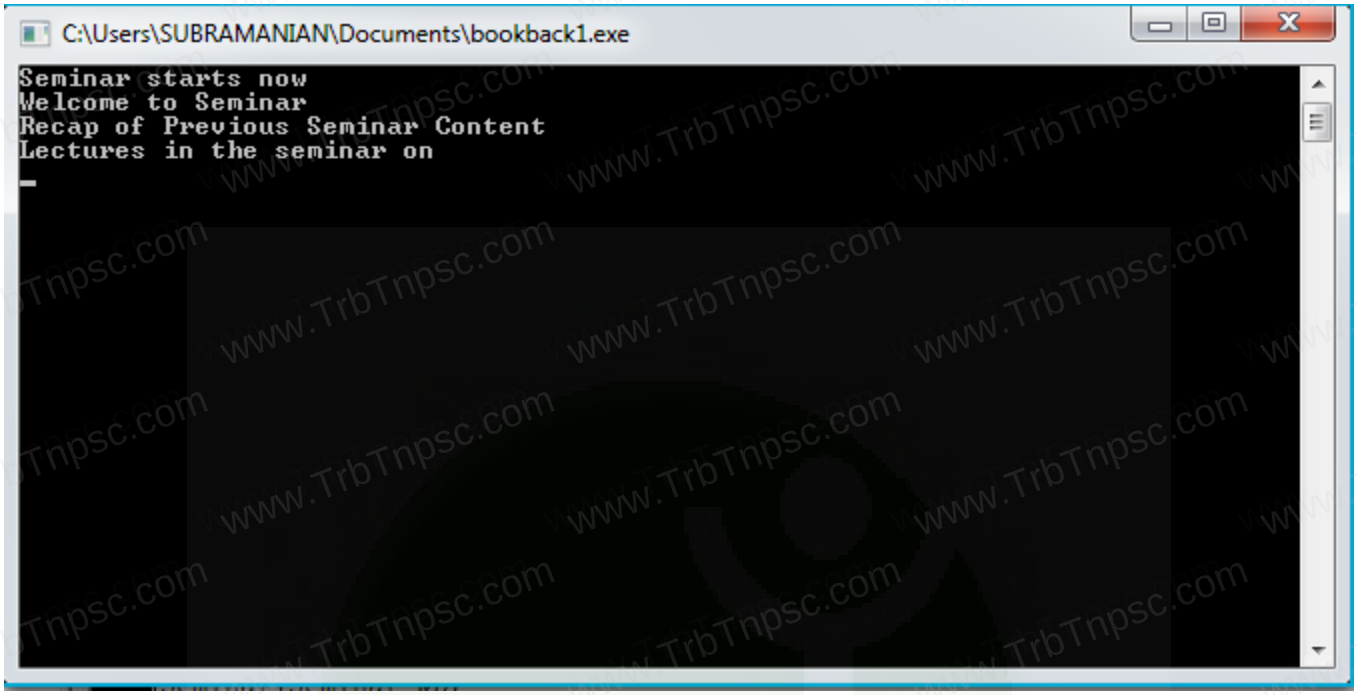
```
Book::Book(int SC,char S[],float f)
{
Bookcode=sc;
strcpy(Bookname,s);
fees=f;
}
```

3. Write the output of the following program

```
include<iostream>
using namespace std;
class Seminar
{
int Time;
public:
Seminar()
{
Time=30;cout<<"Seminar starts now"<<endl;
}
void Lecture()
{
cout<<"Lectures in the seminar on"<<endl;
}
```

```
Seminar(int Duration)
{
Time=Duration;cout<<"Welcome to Seminar "<<endl;
}
Seminar(Seminar &D)
{
Time=D.Time;cout<<"Recap of Previous Seminar Content "<<endl;
}
~Seminar()
{
cout<<"Vote of thanks"<<endl;
}
};
int main()
{
Seminar s1,s2(2),s3(s2);
s1.Lecture();
return 0;
}
```

OUTPUT:



4. Debug the following program

```
#include<iostream>
using namespace std;
class String
{
public:
charstr[20];
public:
void accept_string
{
cout<<"\n Enter String : ";
cin>>str;
}
display_string()
{
cout<<str;
}
```

```
String operator *(String x) //Concatenating String
{
String s;
strcat(str,str);
strcpy(s.str,str);
goto s;
}
}
int main()
{
String str1, str2, str3;
str1.accept_string();
str2.accept_string();
cout<<"\n\n First String is : ";
str1=display_string();
cout<<"\n\n Second String is : ";
str2.display_string();
str3=str1+str2;
cout>>"\n\n Concatenated String is : ";
str3.display_string();
return 0;
}
```

ANSWER:

S.NO	ERROR STATEMENT	CORRECTED STATEMENT	DESCRIPTION
1.	#include<iostream>	#include<iostream.h>	.h is missing
2.	charstr[20];	char str[20];	Space is needed between data type and variable name
3.	void accept_string	void accept_string()	accept_string() does

			not contain parenthesis
4.	display_string()	void display_string()	display_string() does not contain data type
5.	goto s;	No need	There is no such label found in the program.
6.	}	};	Class is not terminated
7.	str1=display_string();	str1.display_string();	Dot Operator is missing
8.	str3=str1+str2;	str3=str1*str2;	Instead of *(asterisk), +(plus) is given.
9.	cout>>"\n\n Concatenated String is : ";	cout<<"\n\n Concatenated String is : ";	Instead of insertion operator they gave extraction operator.

5. Answer the questions based on the following program

```
#include<iostream>
#include<string.h>
using namespace std;
class comp
{
public:
chars[10];
void getstring(char str[10])
{
strcpy(s,str);
}
void operator==(comp);
```

```
};
void comp::operator==(comp ob)
{
if(strcmp(s,ob.s)==0)
cout<<"\nStrings are Equal";
else
cout<<"\nStrings are not Equal";
}
int main()
{
comp ob, ob1;
char string1[10], string2[10];
cout<<"Enter First String:";
cin>>string1;
ob.getstring(string1);
cout<<"\nEnter Second String:";
cin>>string2;
ob1.getstring(string2);
ob==ob1;
return 0;
}
```

(i) Mention the objects which will have the scope till the end of the program.

⊕ No Objects has the scope till the end of the program.

(ii) Name the object which gets destroyed in between the program

⊕ ob

(iii) Name the operator which is over loaded and write the statement that invokes it.

⊕ = = (is Equal to) is the operator which is overloaded.

⊕ ob==ob1; This statement invokes it.

(iv) Write out the prototype of the overloaded member function

⊕ void operator==(comp);

(v) What types of operands are used for the overloaded operator?

⊕ String Operands are used for the overloaded operator.

(vi) Which constructor will get executed? Write the output of the program

⊕ Default constructor gets executed.

CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

CHAPTER 16

INHERITANCE

SECTION – A

I. CHOOSE THE CORRECT ANSWERS:

1. Which of the following is the process of creating new classes from an existing class?
   a) Polymorphism **b) Inheritance** c) Encapsulation d) super class
2. Which of the following derives a class student from the base class school?
   a) school: student **b) class student : public school**
   c) student : public school d) class school : public student
3. The type of inheritance that reflects the transitive nature is
   a) Single Inheritance b) Multiple Inheritance **c) Multilevel Inheritance** d) Hybrid Inheritance
4. Which visibility mode should be used when you want the features of the base class to be available to the derived class but not to the classes that are derived from the derived class?
   a) Private b) Public **c) Protected** d) All of these
5. Inheritance is process of creating new class from
   a) Base class b) Abstract c) Derived class d) Function
6. A class is derived from a class which is a derived class itself, and then this is referred to as
   a) Multiple Inheritance **b) Multilevel Inheritance** c) Single Inheritance d) Double Inheritance
7. Which amongst the following is executed in the order of inheritance?
   a) Destructor b) Member function **c) Constructor** d) Object
8. Which of the following is true with respect to inheritance?
   a) Private members of base class are inherited to the derived class with private
   b) Private members of base class are not inherited to the derived class with private accessibility
   c) Public members of base class are inherited but not visible to the derived class
   d) Protected members of base class are inherited but not visible to the outside class
9. Based on the following class declaration answer the questions (from9.1 o 9.4)

```
class vehicle
{
int wheels;
public:
void input_data(float,float);
void output_data();
protected:
```

```
int passenger;
};
class heavy_vehicle : protected vehicle
{
int diesel_petrol;
protected:
int load;
protected:
int load;
public:
voidread_data(float,float)
voidwrite_data();
};
class bus: private heavy_vehicle
{
charTicket[20];
public:
void fetch_data(char);
void display_data();
};
};
```

9.1 Which is the base class of the class heavy_vehicle?

a) bus b) heavy_vehicle **c) vehicle** d) both (a) and (c)

9.2 The data member that can be accessed from the function displaydata()

a) passenger b) load **c) Ticket** d) All of these

9.3 The member function that can be accessed by an objects of bus Class is

a) input_data() , b) read_data() ,output_data()write_data()

c) fetch_data(), display_data() d) All of these

9.4 The member function that is inherited as public by Class Bus

a) input_data() , b) read_data() , output_data()write_data()

c) fetch_data(), display_data() d) All of these

```
class x
{
```

```
int a;
public :
x()
{}
};
class y
{
x x1;
public :
y(){}
};
class z : public y,x
{
int b;
public:
z()
{
}
}z1;
```

10 What is the order of constructor for object z1 to be invoked?

a) z , y,x,x b) x,y,z,x c) y,x,x,z **d) x,y,z**

SECTION – B

II. ANSWER TO THE ALL QUSTIONS (2 MARKS):

1. What is inheritance?

- ⊕ **Inheritance** is one of the most important features of **Object Oriented Programming**.
- ⊕ The Mechanism of deriving new classes from an existing class is called **Inheritance**
- ⊕ A class that is used as the basis for inheritance is called a **super** class or **base** class.
- ⊕ A class that inherits from a super class is called a **subclass** or **derived** class

2. What is a base class?

- ⊕ A class that is used as the basis for inheritance is called a **Super Class** or **Base Class** or **Parent Class** or **Existing Class.**

⊕ A class which is used to derive other classes are called **Super Class** or **Base Class** or **Parent Class** or **Existing Class.**

3. Why derived class is called power packed class?

⊕ A class that inherits from a super class is called a **Subclass** or **Derived Class** or **Child Class**

⊕ The derived class is a **power packed** class, as it can add additional **attributes** and **methods** and thus enhance its **functionality**.

4. In what multilevel and multiple inheritance differ though both contains many base class?

MULTIPLE INHERITANCE	MULTILEVEL INHERITANCE
⊕ When a derived class inherits from multiple base classes it is known as **Multiple inheritance.**	⊕ The transitive nature of inheritance is itself reflected by this form of inheritance. When a class is derived from a class which is a derived class – then it is referred to as **Multilevel Inheritance.**
⊕ It is not widely used.	⊕ It is widely used than Multiple Inheritance.

5. What is the difference between public and private visibility mode?

PUBLIC VISIBILITY MODE	PRIVATE VISIBILITY MODE
⊕ When a **base** class is inherited with **public** visibility mode, the **protected** members of the **base** class will be inherited as **"protected"** members of the **derived** class and the **public** members of the **base** class will be inherited as **"public"** members of the **derived** class.	⊕ When a **base** class is inherited with **private** visibility mode the **public** and **protected** members of the base class become **"private"** members of the derived class

SECTION – C

III. ANSWER TO THE ALL QUESTIONS (3 MARKS):

1. What are the points to be noted while deriving a new class?

i. The keyword **class** has to be used

ii. The name of the **derived class** is to be given after the keyword **class**

iii. A **single colon (:)**

iv. The type of derivation (the visibility mode), namely **private, public or protected.** If no visibility mode is specified, then by default the visibility mode is considered as private.

v. The names of all base classes (parent classes) separated by **comma**.

2. What is difference between the members present in the private visibility mode and the members present in the public visibility mode?

PUBLIC VISIBILITY MODE	PRIVATE VISIBILITY MODE
⊕ When a **base** class is inherited with **public** visibility mode, the **protected** members of the **base** class will be inherited as **"protected"** members of the **derived** class and the **public** members of the **base** class will be inherited as **"public"** members of the **derived** class.	⊕ When a **base** class is inherited with **private** visibility mode the **public** and **protected** members of the base class become **"private"** members of the derived class

3. What is the difference between polymorphism and inheritance though are used for reusability of code?

POLYMORPHISM	INHERITANCE
⊕ It is basically a common interface of multiple forms.	⊕ It is the newly created class using the properties of the existing class.
⊕ It is implemented only on functions/methods	⊕ It is implemented only in Classes.
⊕ It supports Compile – Time Polymorphism (Overloading) or Run – Time Polymorphism(Overriding)	⊕ It supports the reusability of code and reduces the length of code.
⊕ There are 2 types of Overloading ⊕ Function Overloading, & Operator Overloading.	⊕ There are 5 types of Inheritance.

4. What do you mean by overriding?

⊕ When a derived class member function has the same name as that of its base class member function, the **derived class member function shadows/hides the base class's inherited function.**

⊕ This situation is called **Function Overriding** and this can be resolved by giving the base class name followed by **:: and the member function name.**

5. Write some facts about the execution of constructors and destructors in inheritance

Some Facts About the execution of constructor in inheritance

- ⊕ **Base class** constructors are executed **first**, **before** the **derived class** constructor's execution
- ⊕ **Derived class cannot inherit** the **base class constructor** but it can call the base class constructor by using Base_class name::base_class_constructor() in derived class definition
- ⊕ If there are **Multiple Base classes**, then its start executing from the **left most base class**
- ⊕ In **Multilevel Inheritance**, the **constructors** will be executed in the **order of inheritance**.

SECTION – D

IV. ANSWER TO THE ALL QUESTIONS (5 MARKS):

1. Explain the different types of inheritance?

- ⊕ There are different types of inheritance viz.,
  - ✓ Single Inheritance,
  - ✓ Multiple inheritance,
  - ✓ Multilevel inheritance,
  - ✓ Hybrid inheritance and
  - ✓ Hierarchical inheritance.

SINGLE INHERITANCE

⊕ When a derived class inherits only from one base class, it is known as single inheritance

MULTIPLE INHERITANCE

⊕ When a derived class inherits from multiple base classes it is known as multiple inheritance

HIERARCHICAL INHERITANCE

⊕ When more than one derived classes are created from a single base class, it is known as Hierarchical inheritance.

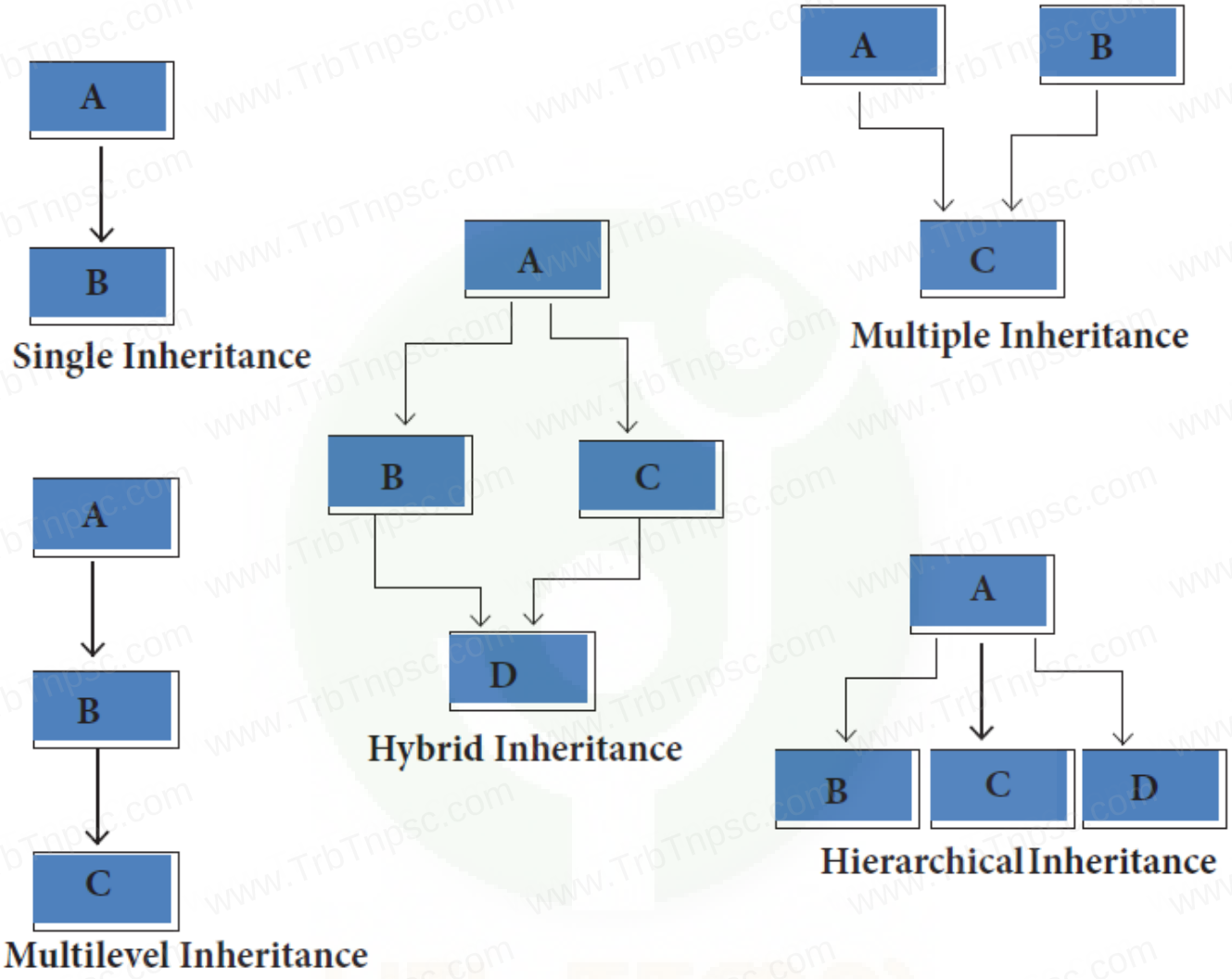
MULTILEVEL INHERITANCE

⊕ The transitive nature of inheritance is itself reflected by this form of inheritance. When a class is derived from a class which is a derived class – then it is referred to as multilevel inheritance.

HYBRID INHERITANCE

⊕ When there is a combination of more than one type of inheritance, it is known as hybrid inheritance.

⊕ Hence, it may be a combination of Multilevel and Multiple inheritance or Hierarchical and Multilevel inheritance or Hierarchical, Multilevel and Multiple inheritance.



2. Explain the different visibility mode through pictorial representation?

⊕ The accessibility of base class by the derived class is controlled by visibility modes.

⊕ The three visibility modes are **Private, Protected and Public.**

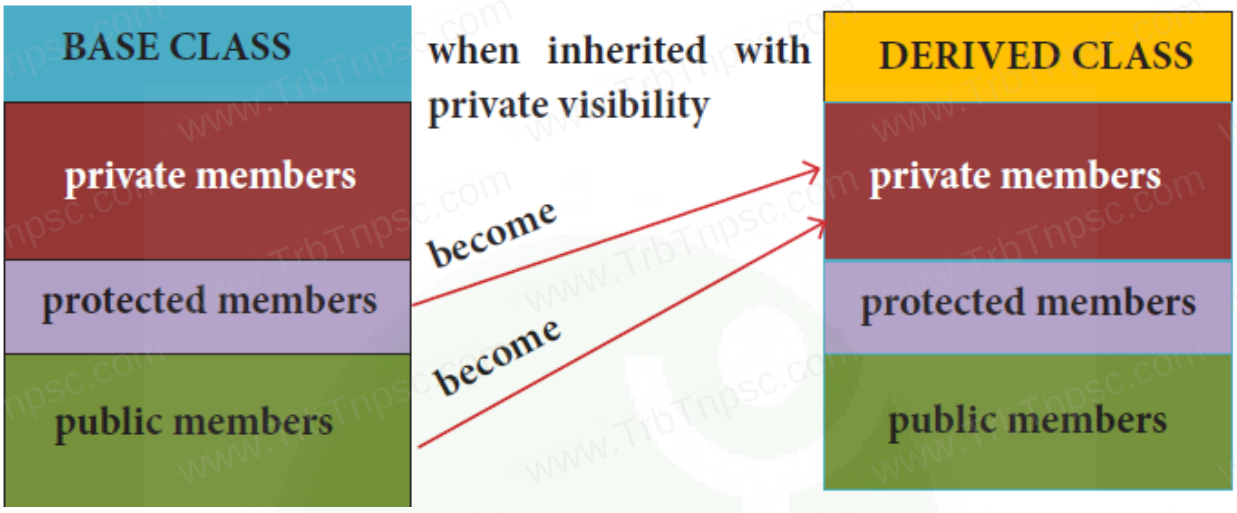
⊕ The **default** visibility mode is **Private.**

⊕ Though visibility modes and access specifiers look similar, the main difference between them is Access specifiers control the accessibility of the members with in the class where as visibility modes control the access of inherited members with in the class.

CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

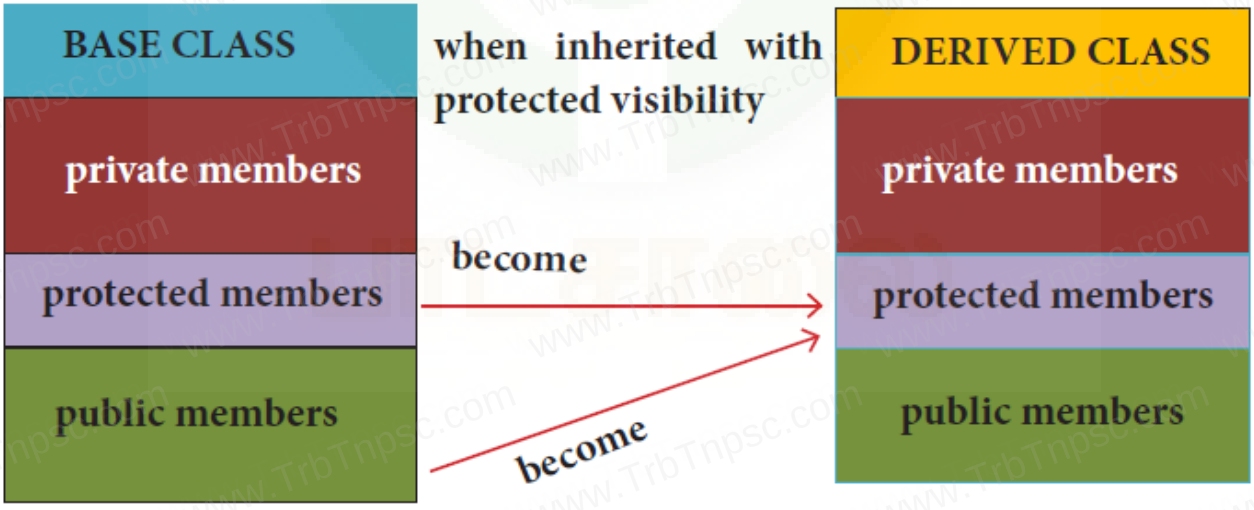
PRIVATE VISIBILITY MODE

⊕ When a base class is inherited with **private** visibility mode the public and protected members of the base class become **'private'** members of the derived class



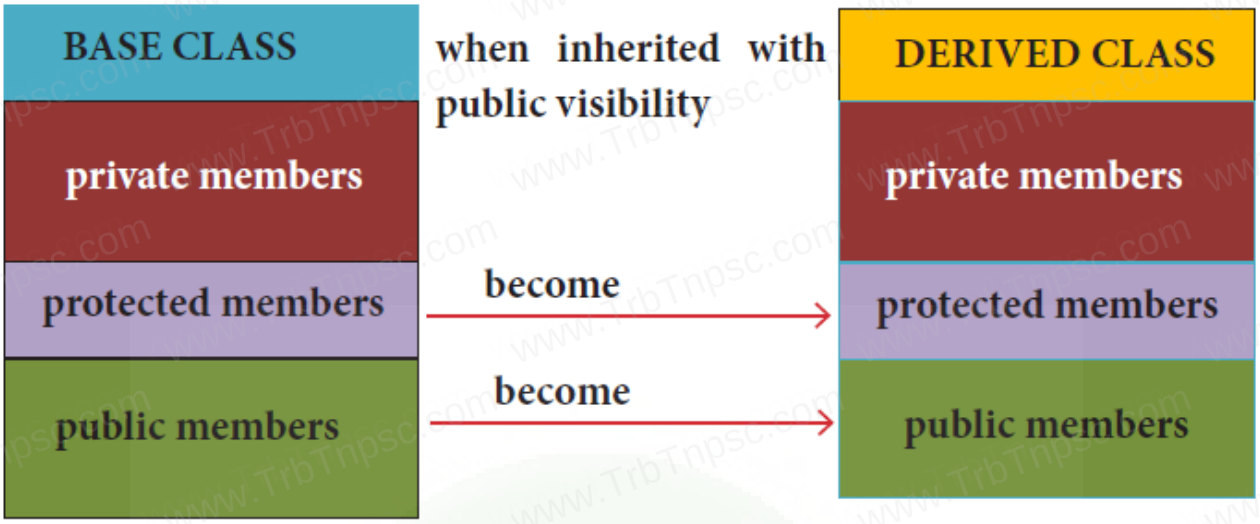
PROTECTED VISIBILITY MODE

⊕ When a base class is inherited with **protected** visibility mode the protected and public members of the base class become **'protected members'** of the derived class



PUBLIC VISIBILITY MODE

⊕ When a base class is inherited with **public** visibility mode, the **protected** members of the **base class** will be inherited as **protected members** of the **derived class** and the **public members** of the **base class** will be inherited as **public members** of the **derived class**.



3. Answer the Questions for the below Program

```
#include<iostream>
#include<string.h>
#include<stdio.h>
using namespace std;
class publisher
{
char pname[15];
char hoffice[15];
char address[25];
double turnover;
protected:
char phone[3][10];
void register();
public:
publisher();
~publisher();
void enterdata();
void dispdata();
};
class branch
{
```

```
char bcity[15];
char baddress[25];
protected:
int no_of_emp;
public:
char bphone[2][10];
branch();
~branch();
void havedata();
void givedata();
};
class author: public branch, publisher
{
int aut_code;
char aname[20];
float income;
public:
author();
~author();
void getdata();
void putdata();
};
```

Answer the following questions based on the above given program:

3.1 Which type of Inheritance is shown in the program?

Multiple Inheritance

3.2 Specify the visibility mode of base classes.

Public

3.3 Give the sequence of Constructor/Destructor Invocation when object of class author is created.

Constructor of base class publisher

Constructor of base class branch

Constructor of derived class author

Destructor of derived class author

Destructor of base class branch

Destructor of base class publisher

3.4 Name the base class(/es) and derived class (/es).

Publisher, Branch is the **Base Classes**

Author is the **Derived Class**

3.5 Give number of bytes to be occupied by the object of the following class:

(a) publisher – **95 bytes of Memory**

(b) branch – **64 bytes of Memory**

(c) author – **28 bytes of Memory**

3.6 Write the names of data members accessible from the object of class author.

phone[3][10];

no_of_emp;

bphone[2][10];

3.7 Write the names of all member functions accessible from the object of class author.

enterdata();

dispdata();
havedata();
givedata();
getdata();
putdata();

3.8 Write the names of all members accessible from member functions of class author.

Data Members:

income

aname[20];

aut_code;

no_of_emp;

phone[3][10];

Member Functions:

getdata();

```
putdata();
havedata();
givedata();
register();
enterdata();
dispdata();
```

4. Consider the following C++ code and answer the questions

```
class Personal
{
int Class,Rno;
char Section;
protected:
char Name[20];
public:
personal();
void pentry();
voidPdisplay();
};
class Marks:private Personal
{
float M{5};
protected:
char Grade[5];
public:
Marks();
void M entry();
void M display();
};
class Result:public Marks
{
float Total,Agg;
public:
```

```
char FinalGrade, Commence[20];
Result();
void R calculate();
void R display();
};
```

Which type of Inheritance is shown in the program?

- ⊕ Multi-level Inheritance is shown in the program

Specify the visibility mode of base classes.

- ⊕ Private, Protected, & Public

Give the sequence of Constructor/Destructor Invocation when object of class Result is created.

Name the base class (/es) and derived class (/es).

- ⊕ Personal is the Base Class for Marks & Marks is the Derived Class
- ⊕ Marks is the Derived Class for Result & Result is the Derived Class

Give number of bytes to be occupied by the object of the following class:

(a) Personal – 25 bytes of Memory
(b) Marks – 10 bytes of Memory
(c) Result – 8 bytes of Memory

Write the names of data members accessible from the object of class Result.

Write the names of all member functions accessible from the object of class Result.

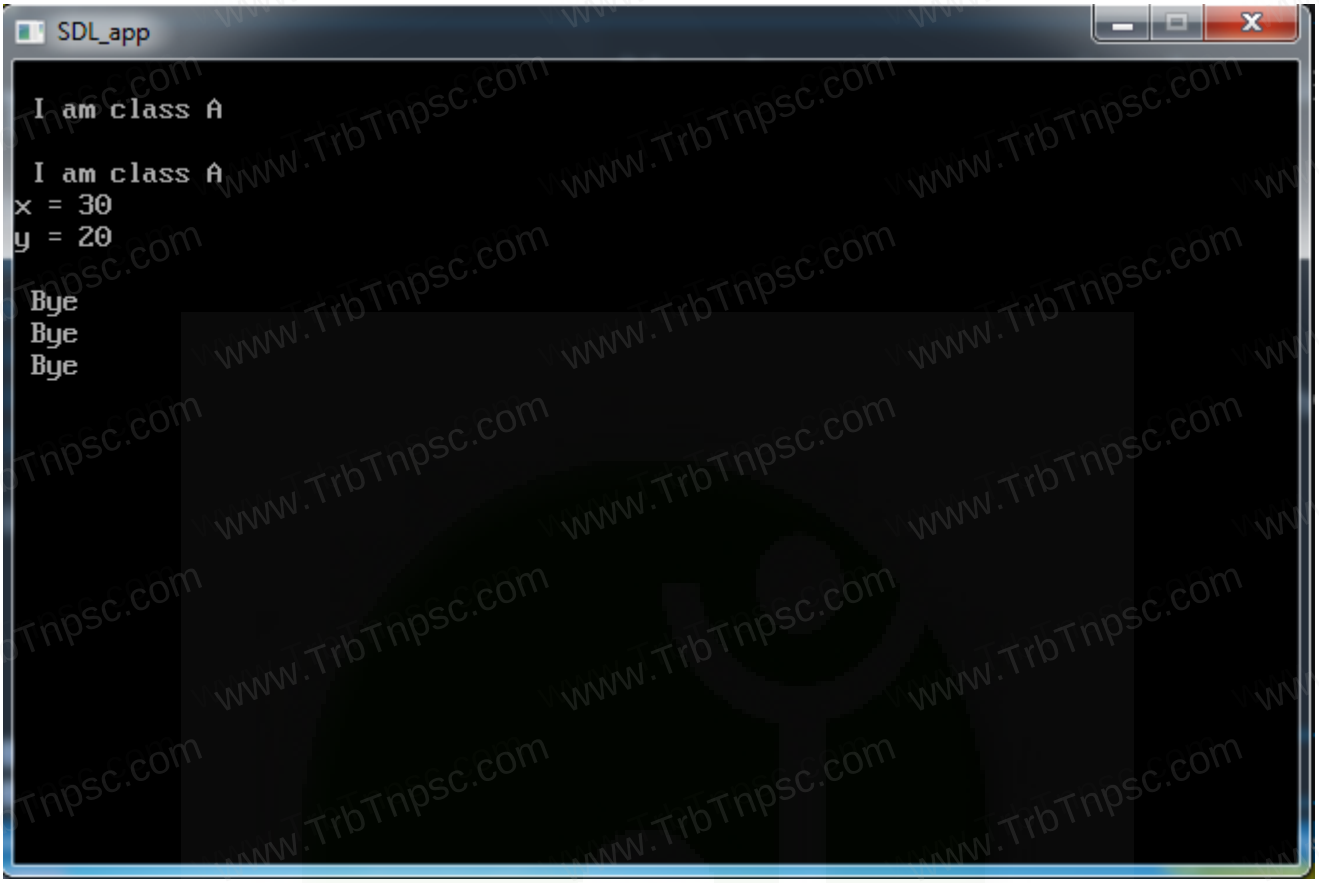
Write the names of all members accessible from member functions of class Result.

5. Write the output of the following program

```
#include<iostream>
using namespace std;
class A
{
protected:
int x;
public:
void show()
{
cout<<"x = "<<x<<endl;
}
A()
{
cout<<endl<<" I am class A "<<endl;
}
~A()
{
cout<<endl<<" Bye ";
}
};
class B : public A
{
{
protected:
int y;
public:
B(int x, int y)
{
this->x = x; //this -> is used to denote the objects datamember
this->y = y; //this -> is used to denote the objects datamember
}
B()
```

```
{
cout<<endl<<" I am class B "<<endl;
}
~B()
{
cout<<endl<<" Bye ";
}
void show()
{
cout<<"x = "<<x<<endl;
cout<<"y = "<<y<<endl;
}
};
int main()
{
A objA;
B objB(30, 20);
objB.show();
return 0;
}
```

OUTPUT:



6. **Debug the following program**

OUTPUT

15

14

13

PROGRAM:

%include(iostream.h)

#include<conio.h>

Class A

{

public;

int a1,a2:a3;

Void getdata[]

{

```
a1=15;
a2=13;a3=13;
}
}
Class B:: public A()
{
PUBLIC
voidfunc()
{
int b1:b2:b3;
A::getdata[];
b1=a1;
b2=a2;
a3=a3;
cout<<b1<<'\t'<<b2<<'t\'<<b3;
}
void main()
{
clrscr()
B der;
der1:func();
getch();
}
```

ANSWER:

S.NO	ERROR STATEMENT	CORRECTED STATEMENT	DESCRIPTION
1.	%include(iostream.h)	#include<iostream.h>	Header file should start with #
2.	Class A	class A	Keyword class should be in lower case
3.	public;	public:	Access Specifier should contain : (Colon)
4.	int a1,a2:a3;	int a1,a2,a3;	Every variable should be separated by comma

CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

5.	Void getdata[]	void getdata()	Keyword should be in lower case. Function contain ( )
6.	Class B:: public A()	class b::public A()	Keyword should be in lower case.
7.	Class B:: public A()	class b:public A()	Inheritance symbol is : (colon)
8.	Class B:: public A()	class b:public A	Class A should not contain ( )
9.	PUBLIC	public:	Access Specifier should contain : (Colon)
10.	voidfunc()	void func()	Space should be allowed
11.	int b1:b2:b3;	int b1,b2,b3;	Every variable should be separated by comma
12.	A::getdata[];	void A::getdata()	Data type is missing
13.	a3=a3;	b3=a3;	Wrong Assignment
14.	clrscr()	clrscr();	Every statement should be terminated
15.	der1:func();	der.func();	Object name is wrong. Dot operator is used to access the member function.

CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

CHAPTER 17

COMPUTER ETHICS AND CYBER SECURITY

SECTION – A

I. CHOOSE THE BEST ANSWER:

1. Which of the following deals with procedures, practices and values?
   a) Piracy b) Programs c) Virus **d) Computer ethics**
2. Commercial programs made available to the public illegally are known as
   a) Freeware **b) Warez** c) Free software d) Software
3. Which one of the following are self-repeating and do not require a computer program to attach themselves?
   a) Viruses **b) Worms** c) Spyware d) Trojans
4. Which one of the following tracks a user visits a website?
   a) Spyware **b) Cookies** c) Worms d) Trojans
5. Which of the following is not a malicious program on computer systems?
   a) Worms **b) Trojans** c) Spyware d) Cookies
6. A computer network security that monitors and controls incoming and outgoing traffic is
   a) Cookies b) Virus **c) Firewall** d) Worms
7. The process of converting cipher text to plain text is called
   a) Encryption **b) Decryption** c) Key d) Proxy server
8. E-commerce means
   a) Electronic commerce b) Electronic data exchange
   c) Electric data exchange d) Electronic commercialization.
9. Distributing unwanted e-mail to others is called.
   a) Scam **b) Spam** c) Fraud d) Spoofing
10. Legal recognition for transactions are carried out by
    a) Electronic Data Interchange b) Electronic Data Exchange
    c) Electronic Data Transfer d) Electrical Data Interchange

SECTION – B

II. ANSWER TO ALL THE QUESTIONS (2 MARKS):

1. What is harvesting?

⊕ A Person or Program collects **login** and **password information** from a legitimate user to illegally gain access to other's account.

2. What are Warez?

- ⊕ Shareware publishers encourage users to give copies of programs to friends and colleagues but ask everyone who uses that program regularly to pay a registration fee to the program's author directly.
- ⊕ **Commercial programs that are made available to the public illegally are often called warez.**

3. Write a short note on cracking.

- ⊕ **Cracking** is where someone edits a **program source** so that the code can be **exploited** or **modified**.
- ⊕ A cracker (also called a black hat or dark side hacker) is a malicious or criminal hacker.
- ⊕ **"Cracking"** means trying to get into **computer systems in order to steal, corrupt, or illegitimately view data.**
- ⊕ **Software Cracking** is the most often used type of cracking which is nothing but removing the encoded copy protection.

4. Write two types of cyber attacks.

WORMS:

- ⊕ **Worms** are **self- repeating** and do not require a computer program to attach themselves.
- ⊕ **Worms** continually look for vulnerabilities and report back to the author of the worm when weaknesses are discovered.

SPYWARE:

- ⊕ **Spyware** can be installed on the computer automatically when the attachments are open, by **clicking on links or by downloading infected software.**

5. What is a Cookie?

- ⊕ A **Cookie** is a **small piece of data sent from a website and stored on the user's computer memory** (Hard drive) by the user's web browser while the user is browsing internet.
- ⊕ **Cookies** were designed to be a **reliable mechanism for websites to remember useful information to record the user's browsing activity** (including clicking particular buttons, logging in etc.).

CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

SECTION – C

III. ANSWER TO ALL THE QUESTIONS (3 MARKS):

1. What is the role of firewalls?

- ⊕ A **firewall** is a computer network security based system that **monitors** and **controls** incoming and **outgoing network** traffic based on predefined security rules.
- ⊕ A **firewall** commonly establishes a **block between a trusted internal computer network and entrusted computer outside the network.**
- ⊕ They are generally categorized as **Network-based or Host-based**.
- ⊕ **Network based firewalls** are positioned on the **gateway computers of LANs [local area Network], WANs [Wide Area Network] and intranets.**
- ⊕ **Host-based firewalls** are positioned on the **network node** itself. The host-based firewall may be a service as a part of the operating system or an agent application such as endpoint security or protection.

2. Write about encryption and decryption.

- ⊕ **Encryption** and **Decryption** are processes that ensure confidentiality that only authorized persons can access the information.
- ⊕ **Encryption** is the process of **translating** the **plain text** into random and **mangled data** (called **cipher-text**).
- ⊕ **Decryption** is the reverse process of converting the **cipher-text back to plaintext.** Encryption and decryption are done by **Cryptography**.

3. Explain symmetric key encryption.

- ⊕ **Symmetric encryption** is a technique to use the **same key** for both **encryption** and **decryption**.
- ⊕ The main **disadvantage** of the **symmetric key encryption** is that all **authorized persons** involved, have to **exchange the key used to encrypt the data before they can decrypt it.**
- ⊕ If anybody intercepts the key information, they may read all messages.

4. What are the guidelines to be followed by any computer user?

- ⊕ **Honesty:** Users should be truthful while using the internet.
- ⊕ **Confidentiality:** Users should not share any important information with unauthorized people.
- ⊕ **Respect:** Each user should respect the privacy of other users.
- ⊕ **Professionalism:** Each user should maintain professional conduct.

- ⊕ **Obey The Law:** Users should strictly obey the cyber law in computer usage.
- ⊕ **Responsibility:** Each user should take ownership and responsibility for their actions

5. What are ethical issues? Name some.

- ⊕ Some of the common ethical issues are listed below:
  - ✓ Cyber crime
  - ✓ Software Piracy
  - ✓ Unauthorized Access
  - ✓ Hacking
  - ✓ Use of computers to commit fraud
  - ✓ Sabotage in the form of viruses
  - ✓ Making false claims using computers

SECTION – D

IV. ANSWER TO ALL THE QUESTIONS (5 MARKS):

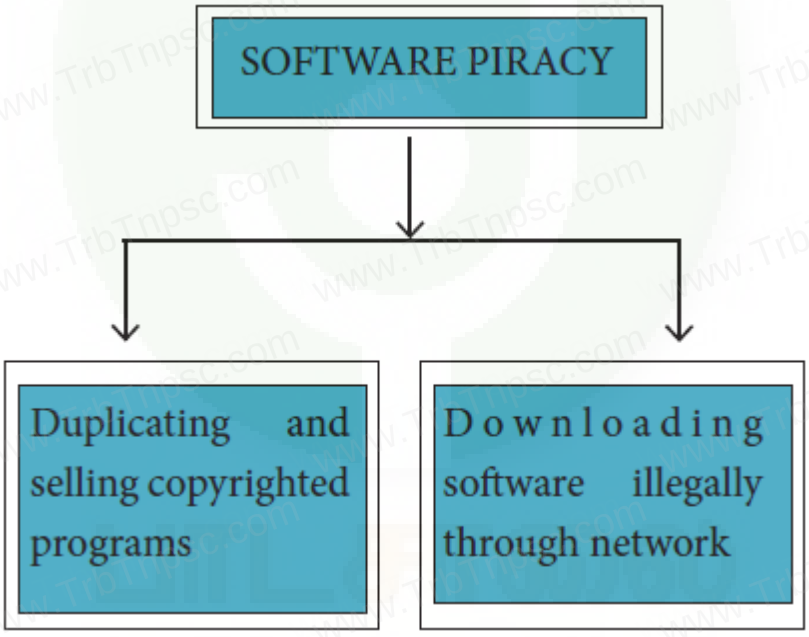
1. What are the various crimes happening using computer?

CRIME	FUNCTION
Crime Function	Hacking, threats, and blackmailing towards a business or a person.
Cyber stalking	Harassing through online.
Malware	Malicious programs that can perform a variety of functions including stealing, encrypting or deleting sensitive data, altering or hijacking core computing functions and monitoring user's computer activity without their permission.
Denial of service attack	Overloading a system with fake requests so that it cannot serve normal legitimate requests.
Fraud	Manipulating data, for example changing the banking records to transfer money to an unauthorized account.
Harvesting	A person or program collects login and password information from a legitimate user to illegally gain access to others' account(s).
Identity theft	It is a crime where the criminals impersonate individuals, usually for financial gain.
Intellectual property theft	Stealing practical or conceptual information developed by another person or company.

Salami slicing	Stealing tiny amounts of money from each transaction.
Scam	Tricking people into believing something that are not true.
Spam	Distribute unwanted e-mail to a large number of internet users.
Spoofing	It is a malicious practice in which communication is send from unknown source disguised as a source known to the receiver.

2. What is piracy? Mention the types of piracy? How can it be prevented?

- ⊕ **Software Piracy** is about the **copyright violation** of **software** created **originally** by an **individual** or an **institution**.
- ⊕ It includes **stealing of codes / programs and other information illegally and creating duplicate copies by unauthorized means and utilizing this data either for one's own benefit or for commercial profit.**
- ⊕ Software Piracy is **"unauthorized copying of software".**



- ⊕ Most of the commercial software is licensed for use at a single computer site or for use by only one user at any time.
- ⊕ When a user buys any software, he becomes a licensed user for that software. He is allowed to make copies of the program for backup purposes, but it is against the law to distribute duplicate copies to others. Such illegal copying and distribution of commercial software should not be practiced.
- ⊕ An entirely different approach to software piracy is called shareware, acknowledges the futility of trying to stop people from copying software and instead relies on people's honesty. Shareware

publishers encourage users to give copies of programs to friends and colleagues but ask everyone who uses that program regularly to pay a registration fee to the program's author directly.

⊕ Commercial programs that are made available to the public illegally are often called warez.

UNAUTHORIZED ACCESS

⊕ Unauthorized access is when someone gains access to a website, program, server, service, or other system by breaking into a legitimate user account.

⊕ For example, if someone tries guessing a password or username for an account that was not theirs until they gained access, it is considered an unauthorized access.

⊕ To prevent unauthorized access, Firewalls, **Intrusion Detection Systems** (IDS), Virus and Content Scanners, Patches and Hot fixes are used.

3. Write the different types of cyber attacks.

⊕ Cyber attacks are launched primarily for causing significant damage to a computer system or for stealing important information from an individual or from an organization.

TYPES OF CYBER ATTACKS:

⊕ Malware is a type of software designed through which the criminals gain illegal access to software and cause damage.

S. NO.	CYBER ATTACK	FUNCTION
1.	**Virus**	A virus is a small piece of computer code that can repeat itself and spreads from one computer to another by attaching itself to another computer file. One of the most common virus is **Trojan**. A Trojan virus is a program that appears to perform one function (for example, virus removal) but actually performs malicious activity when executed.
2.	**Worms**	Worms are self- repeating and do not require a computer program to attach themselves. Worms continually look for vulnerabilities and report back to the author of the worm when weaknesses are discovered.
3.	**Spyware**	Spyware can be installed on the computer automatically when the attachments are open, by clicking on links or by downloading infected software.

4.	**Ransom ware**	Ransom ware is a type of malicious program that demands payment after launching a cyber-attack on a computer system. This type of malware has become increasingly popular among criminals and costs the organizations millions each year.

CHRIST THE KING BOYS MATRIC HR. SEC. SCHOOL, KUMBAKONAM – 612 001.

CHAPTER – 18

TAMIL COMPUTING

SECTION – B

II. ANSWER TO THE FOLLOWING QUESTIONS

1. List of the search engines supporting Tamil.

⊕ In the top ten search engines,

- ✓ **Google,**
- ✓ **Bing and**
- ✓ **Yahoo** are takes first three places respectively

2. What is the keyboard layouts used in Android?

⊕ **Sellinam and Ponmadal** – are familiar Tamil keyboard layouts that work on Android operating system in Smart phone using phonetics.

3. Write a short note about Tamil Programming Language

⊕ Programming languages to develop software to computers and smart phones are available only in English.

⊕ Now, efforts are taken to develop programming languages in Tamil. Based on Python programming language, the first Tamil programming language **"Ezhil"** (**எழில்**) is designed. With the help of this programming language, you can write simple programs in Tamil.

4. What TSCII?

⊕ **TSCII (T**amil **S**cript **C**ode for **I**nformation **I**nterchange**)** is the first coding system to handle our **Tamil language** in an analysis of an encoding scheme that is easily handled in **electronic devices, including non-English computers.**

⊕ This encoding scheme was registered in **IANA** (**I**nternet **A**ssigned **N**umbers **A**uthority) unit of **ICANN.**

5. Write a short note on Tamil Virtual Academy.

⊕ With the objectives of spreading Tamil to the entire world through internet, Tamil Virtual University was established on 17th February 2001 by the Govt. of Tamilnadu.

⊕ This organization is functioning with the name "Tamil Virtual Academy".

⊕ This organization offers different courses regarding Tamil language, Culture, heritage etc., from kindergarten to under graduation level.