UNIT I

Syllabus: Introduction to C++: Difference between C and C++, Evolution of C++, The Object Oriented Technology, Disadvantage of Conventional Programming, Key Concepts of Object Oriented Programming, Advantage of OOP, Object Oriented Language.

1. **Introduction:**

C++ is an object oriented programming language, C++ was developed by Jarney Stroustrup in 1983 at AT & T Bell laboratories, USA. C++ was developed from C and simula 67 language. C++ was early called 'C with classes'.

- C++ is derived from C Language. It is a Superset of C.
- Earlier C++ was known as C with classes.
- In C++, the major change was the addition of classes and a mechanism for inheriting class objects into other classes.
- Most C Programs can be compiled in C++ compiler.
- C++ expressions are the same as C expressions.
- All C operators are valid in C++.

2. Differences between C and C++

Following are the differences Between C and C++ :

S.No.	C	C++
1	C is Procedural Language.	C++ is non Procedural i.e Object oriented Language.
2	No virtual Functions are present in C	The concept of virtual Functions are used in C++.
3	In C, Polymorphism is not possible.	The concept of polymorphism is used in C++. Polymorphism is the most Important Feature of OOPS.

4	Operator overloading is not possible in C.	Operator overloading is one of the greatest Feature of C++.
5	Top down approach is used in Program Design.	Bottom up approach adopted in Program Design.
6	No namespace Feature is present in C Language.	Namespace Feature is present in C++ for avoiding Name collision.
7	Multiple Declaration of global variables are allowed.	Multiple Declaration of global varioables are not allowed.
8	In C • scanf() Function used for Input. • printf() Function used for output.	In C++ • Cin>> Function used for Input. • Cout<< Function used for output.
9	Mapping between Data and Function is difficult and complicated.	Mapping between Data and Function can be used using "Objects"
10	In C, we can call main() Function through other Functions	In C++, we cannot call main() Function through other functions.
11	C requires all the variables to be defined at the starting of a scope.	C++ allows the declaration of variable anywhere in the scope i.e at time of its First use.
12	No inheritance is possible in C.	Inheritance is possible in C++
13	In C, malloc() and calloc() Functions are used for Memory Allocation and free() function for memory Deallocating.	In C++, **new** and **delete** operators are used for Memory Allocating and Deallocating.
14	It supports built-in and primitive data	It support both built-in and user

	types.	define data types.
15	In C, does not provide direct support to Eexception handling.	In C++, Exception Handling is done with Try and Catch block.

3. **Evolution of C++**

C++ is an object oriented programming language and also considered as an extension of C. Bjarne Stroustrup at AT&T Bell Laboratories in Murray Hill, New Jersey (USA) developed this language in the early 1980s. Stroustrup, a master of Simula67 and C, wanted to combine the features of both the languages and he developed a powerful language that supports object-oriented programming with features of C. The outcome was C++ as per Fig. 1.1. Various features were derived from ***SIMULA67*** and ***ALGOL68***. Stroustrup called the new language ***'C with classes'***. However, in 1983, the name was changed to C++.

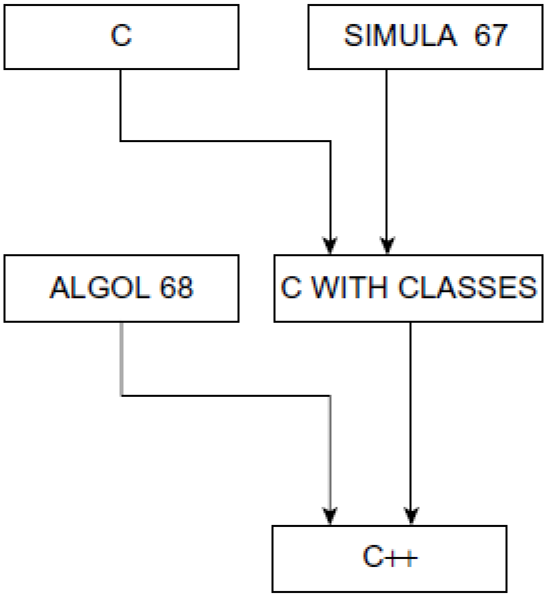


Fig. 1.1 Evolution of C++

The thought of C++ came from the C increment operator ++. Rick Mascitti coined the term C++ in 1983. Therefore, C++ is an extension of C. C++ is a superset of C. All the concepts of C are applicable to C++ also.

For developing complicated applications, object oriented language such as C++ is the most convenient and easy. Hence, a programmer must be aware of its features.

4. THE OBJECT ORIENTED TECHNOLOGY

Nature is composed of various objects. Living beings can be categorized into different objects.

Let us consider an example of a teaching institute which has two different working sections – teaching and non-teaching. Further sub-grouping of teaching and non-teaching can be made for the coordination of management. The various departments of any organization can be thought of as objects working for certain goals and objectives.

Usually an institute has faculty of different departments. The Director/Principal is a must for the overall management of the institute. The Academic Dean is responsible for the academics of the institute. The Dean for Planning should have the future plans of the institute and he/she must see how the infrastructure is utilized effectively. The Dean R&D should see research activities run in the institute forever.

Besides teaching staff there must be laboratory staff for assistance in conducting practical sessions, and a site development section for beautification of the campus. The accounts department is also required for handling monetary transactions and salaries of the employees. The Sports section is entrusted the responsibility of sports activities. The Registrar for Administration and staff for dealing with administrative matters of the institute are also required. Each department has an in-charge who carries clear-cut given responsibilities. Every department has its own work as stated above. When an institute's work is distributed into departments as shown in Fig. 1.2, it is comfortable to accomplish goals and objectives. The activities are carried on smoothly. The burden of one particular department has to be shared among different departments with personnel. The staff in the department is controlled properly and act according to the instructions laid down by the management. The faculty performs activities related to teaching. If the higher authority needs to know the details regarding the theory, practical, seminar and project loads of individuals of the department, then a person from the department furnishes the same to the higher authority. This way some responsible person from the department accesses the data and provides the higher authority with the requisite information. It is also good to think that no unconnected person from another department reads the data or attempts to make any alteration that might corrupt the data.

As shown in Fig. 1.2, an institute is divided into different departments such as library, classroom, computer laboratory, etc. Each department performs its own activities in association with the other departments. Each department may be considered as a module and it contains class and object in C++ language. This theory of class and object can be extended to every walk of life and can be implemented with software. In general, objects are in terms of entities.

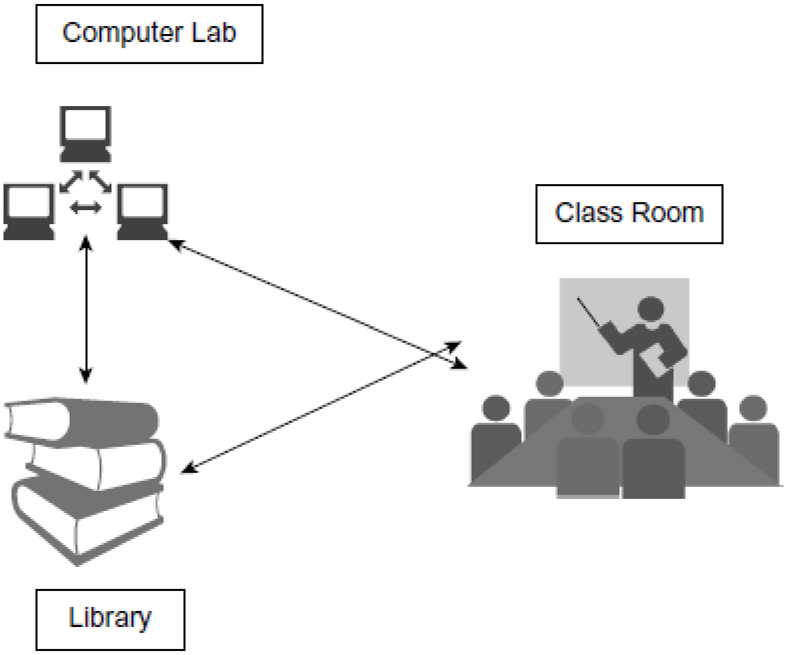


Fig.1.2 Relationship between different sections

In a nutshell, in object oriented programming objects of a program interact by sending messages to each other.

5. DISADVANTAGE OF CONVENTIONAL PROGRAMMING

Traditional programming languages such as COBOL, FORTRAN, C etc. are commonly known as procedure oriented languages. The program written in these languages consists of a sequence of instructions that tells the compiler or interpreter to perform a given task. Numerous functions are initiated by the user to perform a task. When a program code is large, it becomes inconvenient to manage it. To overcome this problem, procedures or subroutines were adopted to make a program more understandable to the programmers. A program is divided into many functions.

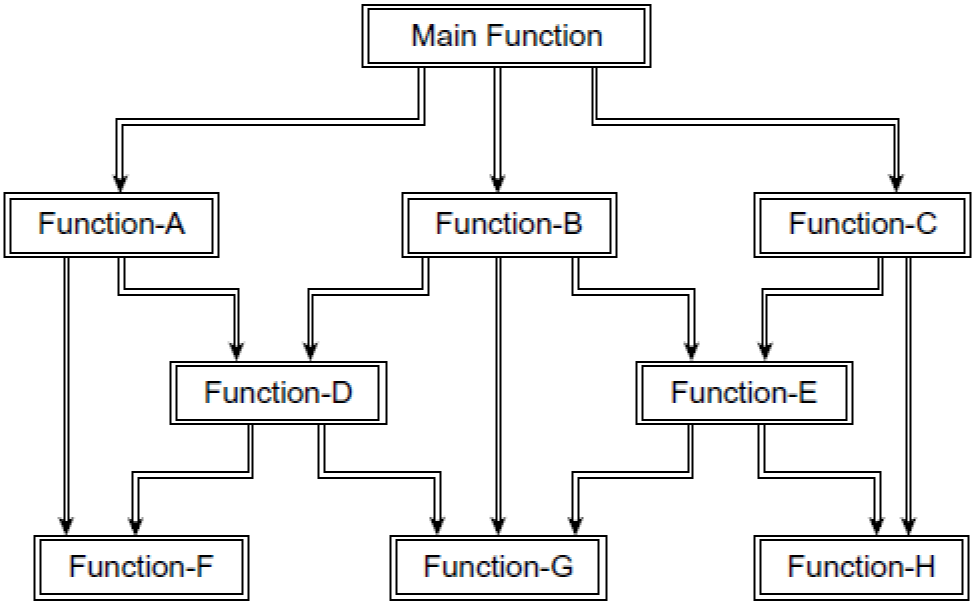


Fig. 1.3 Flow of functions in non-OOP languages

Each function can call another function, as shown in Fig. 1.3. Each function has its own task. If the program is too large the function also creates problems. In many programs, important data variables are declared as global. In case of programs containing several functions, every function can access the global data as per the simulation in Fig. 1.4. In huge programs it is difficult to know what data is used by which function. Due to this the program may contain several logical errors.

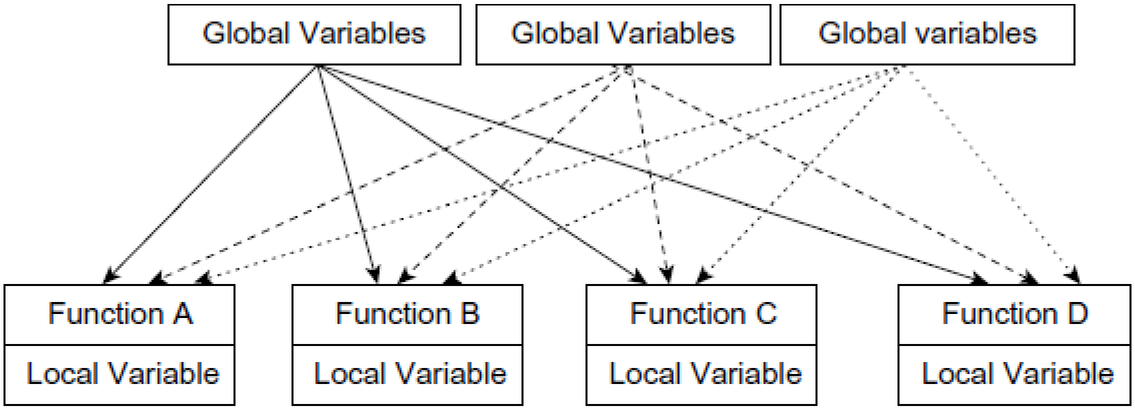


Fig. 1.4 Sharing of data by functions in non-OOP languages

The following are the drawbacks observed in monolithic, procedure, and structured programming languages:

1. Huge programs are divided into smaller programs known as functions. These functions can call one another. Hence security is not provided.
2. No importance is given to security of data and importance is laid on doing things.
3. Data passes globally from function to function.
4. Most function accesses global data.

6.Key Concepts of Object Oriented Programming

The major Concepts of Object Oriented Programming are:

1. Class
2. Object
3. Abstraction
4. Encapsulation
5. Data Hiding
6. Inheritance
7. Reusability
8. Polymorphism
9. Virtual Functions
10. Message passing

Class: Class is an abstract data type (user defined data type) that contains member variables and member functions that operate on data. It starts with the keyword class. A class denotes a group of similar objects.

```
e.g.:  class employee

          {

           int empno;

           char name[25],desg[25];

           float sal;

          public:
```

```
void getdata ();

void putdata ();

 };
```

Object: An object is an instance of a class. It is a variable that represents data as well as functions required for operating on the data. They interact with private data and functions through public functions.

```
e.g.:        employee e1, e2;
```

In the above example employee is the class name and e1 and e2 are objects of that class.

Abstraction: Abstraction refers to the process of concentrating on the most essential features and ignoring the details. There are two types of abstraction

i) Procedural Abstraction
ii) Data Abstraction

Procedural Abstraction: Procedural abstraction refers to the process of using user-defined functions or library functions to perform a certain task, without knowing the inner details. The function should be treated as a black box. The details of the body of the function are hidden from the user.

Data Abstraction: Data Abstraction refers to the process of formation of user defined data type from different predefined data types.

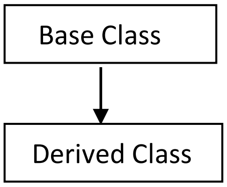
e.g: structure, class.

Encapsulation: Encapsulation is the process of combining data members and member functions into a single unit as a class in order to hide the internal operations of the class and to support abstraction.

Data Hiding: All the data in a class can be restricted from using it by giving some access levels (visibility modes). The three access levels are private, public, protected.

Private data and functions are available to the public functions only. They cannot be accessed by the other part of the program. This process of hiding private data and functions from the other part of the program is called as data hiding.

Inheritance: Inheritance is the process of acquiring (getting) the properties of some other class. The class whose properties are being inherited is called as base class and the class which is getting the properties is called as derived class.



Reusability: Using the already existing code is called as reusability. This is mostly used in inheritance. The already existing code is inherited to the new class. It saves a lot of time and effort. It also reduces the size of the program.

Polymorphism: Polymorphism means the ability to take many forms. Polymorphism allows to take different implementations for same name.

poly → many

morphism → forms

There are two types of polymorphism, Compile time polymorphism and run time polymorphism.

In Compile time polymorphism binding is done at compile time and in runtime polymorphism binding is done at runtime.

e.g.: Function overloading, operator overloading

Function Overloading: Function overloading is a part of polymorphism. Same function name having different implementations with different number and type of arguments.

Operator Overloading: Operator overloading is a part of polymorphism. Same operator can have different implementations with different data types.

Virtual Functions: Virtual functions are special type of functions which are defined in the base class and are redefined in the derived class. When virtual function is called with a base pointer and derived object then the derived class function will be called. A function can be defined as virtual by placing the keyword virtual for the member function.

Message Passing: An object-oriented program contains a set of objects that communicate with one another. The process of object oriented programming contains the basic steps:

1. Creating classes
2. Creating objects
3. Communication among objects

This communication is done with the help of functions (i.e., passing objects to functions)

7.Advantage of OOP

Object oriented technology provides many advantages to the programmer and the user. This technology solves many problems related to software development, provides improved quality and low cost software.

1. Object oriented programs can be comfortably upgraded.
2. Using inheritance, we can eliminate redundant program code and continue the use of previously defined classes.

3. The technology of data hiding facilitates the programmer to design and develop safe programs that do not disturb code in other parts of the program.
4. The encapsulation feature provided by OOP languages allows programmer to define the class with many functions and characteristics and only few functions are exposed to the user.
5. All object oriented programming languages allows creating extended and reusable parts of programs.
6. Object oriented programming changes the way of thinking of a programmer. This results in rapid development of new software in a short time.
7. Objects communicate with each other and pass messages.

Applications of OOPS

The promising areas for application of OOP includes:

- Real-time systems
- Simulation and modeling
- Object-oriented databases
- Hypertext, hypermedia and experttext
- AI and expertsystems
- Neural networks and parallel programming
- Decision support and Automation system
- CIM/CAM/CAD systems

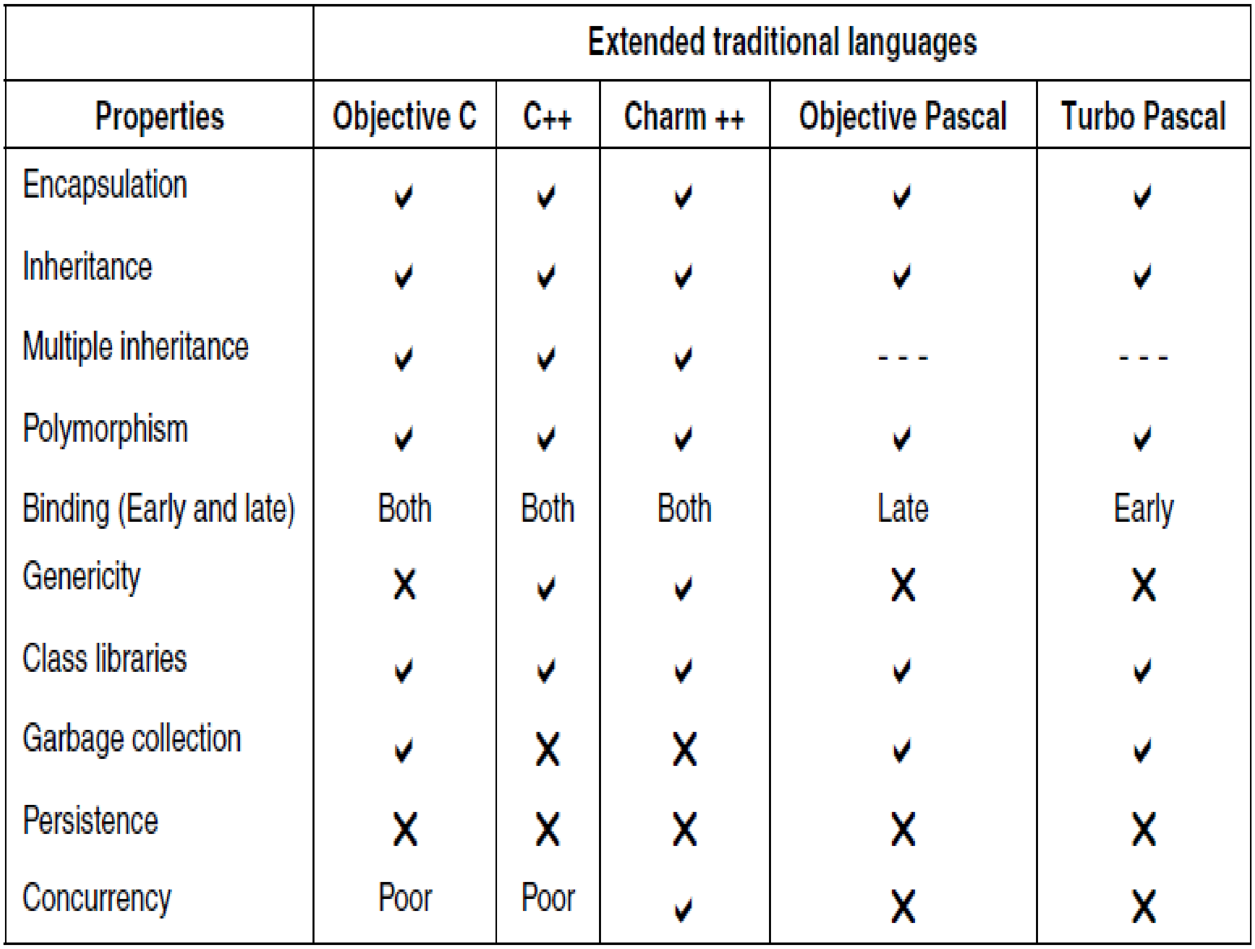
8.Object Oriented Languages

There are many languages which support object oriented programming. Tables 1.1 and 1.2 describe the OOP languages and features supported by them.

Table 1.1 Properties of pure OOP and object based languages

	Pure Object Oriented Languages					Object Based Languages
Properties	**Java**	**Simula**	**Smalltalk**	**Eiffel**	**Java**	**Ada**
Encapsulation	✓	✓	✓	✓	✓	✓
Inheritance	✓	✓	✓	✓	✓	No
Multiple inheritance	✗	✗	✓	✓	✗	No
Polymorphism	✓	✓	✓	✓	✓	✓
Binding (Early and late)	Both	Both	Late binding	Early binding	Both	Early binding
Genericity	✗	✗	✗	✓	✗	✓
Class libraries	✓	✓	✓	✓	✓	Few
Garbage collection	✓	✓	✓	✓	✓	✗
Persistence	✓	✗	Promised	Less	✓	Same as 3GL
Concurrency	✓	✓	Less	Promised	✓	Hard

Table 1.2 Properties of extended traditional languages



	Extended traditional languages				
Properties	Objective C	C++	Charm ++	Objective Pascal	Turbo Pascal
Encapsulation	✓	✓	✓	✓	✓
Inheritance	✓	✓	✓	✓	✓
Multiple inheritance	✓	✓	✓	- - -	- - -
Polymorphism	✓	✓	✓	✓	✓
Binding (Early and late)	Both	Both	Both	Late	Early
Genericity	X	✓	✓	X	X
Class libraries	✓	✓	✓	✓	✓
Garbage collection	✓	X	X	✓	✓
Persistence	X	X	X	X	X
Concurrency	Poor	Poor	✓	X	X

The following are the object-oriented languages, which are widely accepted by the programmer.

- C++
- Smalltalk
- Charm ++
- Java

SMALLTALK

Smalltalk is a pure object oriented language. C++ makes few compromises to ensure quick performance and small code size. Smalltalk uses run-time binding. Smalltalk programs are considered to be faster than the C++. Smalltalk needs longer time to learn than C++. Smalltalk programs are written using Smalltalk browser. Smalltalk uses dynamic objects and memory is allocated from free store. It also provides automatic garbage collection and memory is released when object is no longer in use.

CHARM++

Charm ++ is also an object oriented programming language. It is a portable. The language provides features such as inheritance, strict type checking, overloading, and reusability. It is designed in order to work efficiently with different parallel systems together with shared memory systems, and networking.

Answer the following questions.

1. Compare C and C++.
2. What is object oriented programming?
3. Explain the key concepts of OOP.
4. What are the disadvantages of conventional programming languages?
5. Explain the characteristics of monolithic programming languages.
6. List the disadvantages of procedural programming languages.
7. Explain evolution of C++.
8. List the names of popular OOP languages.
9. List the unique features of an OOP paradigm.
10. What is an object and class?
11. How is data secured in OOP languages?
12. Compare and contrast OOP languages with procedure oriented languages.
13. Mention the types of relationships between two classes.
14. What is structured oriented programming? Discuss its pros and cons.
15. How is global data shared in procedural programming?
16. Describe any two object oriented programming languages.
17. What are the differences between Java and C++?
18. Mention the advantages of OOP languages?
19. What do you mean by message passing?
20. Distinguish between inheritance and delegation.

Answer the following by selecting the appropriate option.

1. Data hiding concept is supported by language
    1. C
    2. Basic
    3. Fortran
    4. **C++**

1. Function overloading means
    1. different functions with different names
    2. function names are same but same number of arguments
    3. **function names are same but different number of arguments**
    4. none of the above

3. C++ language was invented by
    1. **Bjarne Stroupstrup**
    2. Dennis Ritche
    3. Ken Thompson
    4. none of the above

4. The languages COBOL and BASIC are commonly known as
    1. **procedure oriented languages**
    2. object oriented languages
    3. low level languages
    4. none of the above

5. A program with only one function is observed in
    1. **monolithic programming languages.**
    2. object oriented languages
    3. structured programming languages
    4. none of the above

6. The packing of data and functions into a single component is known as
    1. **encapsulation**
    2. polymorphism
    3. abstraction
    4. none of the above

7. The method by which objects of one class get the properties of objects of another class is known as
    1. **inheritance**
    2. encapsulations
    3. abstraction
    4. none of the above

8. The mechanism that allows same functions to act differently on different classes is known as

1. **polymorphism**
2. encapsulations
3. inheritance
4. none of the above

9.The existing class can be reused by

1. **inheritance**
2. polymorphism
3.dynamic binding
4.abstraction

10.Composition of objects in a class is known as

1. **delegation**
2. inheritance
3.polymorphism
4.none of the above

11.A class

1. binds the data and its related functions together
2.data and their addresses
3.contains only functions
4.none of the above

12. The major drawback of procedural programming languages is

1. frequently invoking functions from `main()`
2. **non security of data**
3. non security of methods
4. none of the above