

DIFFERENCES

Procedural Programming Language	Object Oriented Programming Language(OOP)
1. The procedural programming executes series of procedures sequentially	1. In object oriented programming approach there is a collection of objects.
2. This is a top down programming approach.	2. This is a bottom up programming approach.
3. The major focus is on procedures of functions.	3. The main focus is on objects.
4. Data reusability is not possible.	4. Data reusability is one of the important feature of OOP.
5. Data hiding is not possible.	5. Data hiding can be done by making it private.
6. It is simple to implement.	6. It is complex to implement.
For example: C, Fortran, and COBOL.	For example: C++, JAVA.

Class	Object
1. For a single class there can be any number of objects. For example- if we define the class as River then Ganga, Yamuna, Narmada can be the objects of the class River.	1. There are many objects that can be created from one class. These objects make use of the methods and attributes defined by the belonging class.
2. The scope of the class is persistent throughout the program.	2. The objects can be created and destroyed as per the requirements.
3. The class cannot be initialized with some property values.	3. We can assign some property values to the objects.
4. A class has unique name.	4. Various objects having different names can be created for the same class.

Data encapsulation	Data abstraction
1. It is a process of binding data members of a class to the member functions of that class.	1. It is the process of eliminating unimportant details of a class. In this process only important properties are highlighted.
2. Data encapsulation depends upon object data type.	2. Data abstraction is independent upon object data type.
3. It is used in software implementation phase.	3. It is used in software design phase.
4. Data encapsulation can be achieved by inheritance.	4. Data abstraction is represented by using abstract classes.

Differences

Inheritance	Polymorphism
1. Inheritance is a property in which some of the properties and methods of base class can be derived by the derived class.	1. Polymorphism is ability for an object to used different forms. The name of the function remains the same but it can perform different tasks.
2. Various types of inheritance can be single inheritance, multiple inheritance, multilevel inheritance and hybrid inheritance.	2. Various types of polymorphism are compile time polymorphism and run time polymorphism. In compile time polymorphism there are two types of overloading possible. – Functional overloading and operator overloading. In run tie polymorphism there is a use of virtual function.

C	C++	JAVA
1. The C language needs to be compiled.	1. The C++ is a language that needs to be compiled.	1. The Java is a language that gets interpreted and compiled.
2. The C is platform dependant.	2. C++ is platform dependant.	2. Java is a platform independent.
3. There is no concept of threading in C.	3. C++ does not support multi-threading programming	3. Java supports multi-threading.
4. Using C, the GUI applications cannot be created.	4. C++ does not have facility to create and implement the graphical user interface.	4. Using Java, one can design very interactive and user friendly graphical user interface.
5. C does not support database oriented application.	5. Database handling using C++ is very complex. The C++ does not allow the Database connectivity.	5. Java servlets can be created which can interact with the database like MS-ACCESS, Oracle, My-SQL and so on.
6. C cannot be embedded in scripting language.	6. C++ code cannot be embedded in any scripting language.	6. Java code can be embedded within a scripting language by mean of Applet Programming.
7. There is no concept of inheritance.	7. C++ supports multiple inheritance.	7. Java does not support multiple inheritance however it make use of interface.
8. C uses pointers.	8. In C++ we can use the pointers.	8. In Java there is no concept of pointers.
9. C does not support template.	9. C++ supports template.	9. Java does not support the concept of templates.
10. There is no class in C.	10. In C++ we can write a program without a class.	10. In Java, there must be at least one class present.

Differences

11. C is the most simple.	11. C++ is simple to use and implement.	11. Java is safe and more reliable.
12. C can be compiled on variety of compilers.	12. C++ can be compiled with variety of compilers.	12. Java can be compiled using an unique compiler.

Method Overloading	Method Overriding
1. The method overloading occurs at compile time.	1. The method overriding occurs at the run time or execution time.
2. In case of method overloading different number of parameters can be passed to the function.	2. In function overriding the number of parameter that are passed to the function are the same.
3. The overloaded functions may have different return types.	3. In method overriding all the methods will have the same return types.
4. Method overloading is performed within a class.	4. Method overriding is normally performed between two classes that have inheritance relationship.

Arrays	Vectors
1. Array contains all the elements of similar data type.	1. Vector contains the elements that may be of varying data type.
2. The size of array is fixed.	2. The size of array is varying. We can change the size of the vector whenever is required.
3. We can store the simple elements directly in the arrays.	3. We cannot store the simple objects in the database.
4. We can directly add the simple data type element in the array.	4. We cannot directly add the simple data type element in the vector we need to add an object.

Class	Interface
1. The class is denoted by a keyword class.	1. The interface is denoted by a keyword.
2. The class contains data members and methods. But the methods are defined in class implementation. Thus class contains an executable code.	2. The interface may contain data members and methods but the methods are not defined. The Interface serves as an outline for the class.
3. By creating an instance of a class the class members can be accessed.	3. You cannot create an instance of an interface.
4. The class contains data members and methods. But the methods are defined in	4. The interfaces may contains data members and methods but the methods are not defined. The

Differences

class implementation. Thus class contains an executable code.	interface serves as an outline for the class.
5. By creating an instance of a class the class members can be accessed.	5. You can create an instance of an interface.

Thread	Process
1. Thread is a light-weight process.	1. Process is a heavy weight process.
2. Threads do not require separate address space for its execution. It runs in address space of the process to which it belongs to.	2. Each process requires separate address space to execute.

Multithreading	Multitasking
1. Thread is a fundamental unit of multithreading.	1. Program or Process is a fundamental unit of multitasking environment.
2. Multiple parts of a single program gets executed is multithreading environment.	2. During multitasking the processor switches between multiple programs or processes.
3. During multithreading the processor switches between multiple threads of the program.	3. During multitasking the processor switches between multiple programs or processes.
4. It is cost effective because CPU can be shared among multiple threads at a time.	4. It is expensive because when a particular process uses CPU other processes has to wait.
5. It is highly efficient.	5. It is less efficient in comparison with multithreading.
6. It helps in developing efficient applications programs.	6. It helps in developing efficient operating system programs.