Differences Between Object Oriented Programming Languages (C++, Java, C#)

Student : Vo Tran Thanh Luong
Student code : 1551020

I. Introduction :

Nowadays, there are a lot of OOP languages.

The first is C++. C++ is a general-purpose programming language, which introduces object-oriented programming features to C. It offers classes, which provide the four features commonly present in OOP languages: abstraction, encapsulation, inheritance, and polymorphism.

Then comes Java. Java is a general-purpose computer programming language that is heavily inspired from C and C++. However they are not the same. Java is concurrent, class-based, object-oriented, and is intended to let application developers "write once, run anywhere" meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

The latest created is C#. The C# language is intended to be a simple, modern, general-purpose, object-oriented programming language. It is built by Microsoft to improve some aspects of Java.

It is of top important to know some specific differences among these three languages and how they implemented OOP paradigm.

II. Detailed Comparision :

	C++	Java	C#
Class and Object. Inheritance and Polymorphism	1. Object can be created on stack or wherever possible 2. We can have free floating functions. 3. Support mutiple inheritance. 4. Support friend function 5. In C++ methods are non-virtual by default, so to replace the behaviour (allow over-riding) of a method in the derived class you have to explicitly use the virtual keyword in the parent class. 6. abstract class can be achieved using virtual keyword. There is little difference between abstract class and interface. 7. C++ supports pointers, even to classes	1. Objects are created on heap. 2. Functions are parts of class. 3. Do not support mutiple inheritance, but replace it by interface 4. No support for friend functions. You put your friends in the same “package” 5. In Java, methods are virtual by default and we always operate on the dynamic type of the object. You cannot specify a method as non-virtual, but there is a final keyword. Once you use the final keyword on a method you cannot replace it - so there are no issues with static and dynamic types 6. In Java, methods are virtual by default and we always operate on the dynamic type of the object. You cannot specify a method as non-virtual, but there is a final keyword. Once you use the final keyword on a method you cannot replace it - so there are no issues with static and dynamic types 7. There are differences between abstract classes and interfaces 8. Java doesn’t support pointer	1. Objects are created on heap. 2. Functions are parts of class. 3. Use the abstract modifier in a method or property declaration to indicate that the method or property does not contain implementation. 4. When a base class declares a method as virtual, a derived class can override the method with its own implementation. If a base class declares a member as abstract, that method must be overridden in any non-abstract class that directly inherits from that class. If a derived class is itself abstract, it inherits abstract members without implementing them 5. Class can be declared as abstract to prevent direct instantiation by using the new keyword. 6. Support interface as a reference type that is somewhat similar to an abstract base class that consists of only abstract members.

Constructor	Pretty much the same in 3 languages	Pretty much the same in 3 languages	Support private constructor (A private constructor is a special instance constructor. It is generally used in classes that contain static members only.)
Destructor	Destrutor is built inside a class	- Java has no destructor	- Variable.dispose() void Dispose() { option }
Operators Overloading	Operator overloading for most operators. Preserving meaning (semantics) is highly recommended.	Operators are not overridable. The language overrides + and += for the String class.	- You can redefine or overload most of the built-in operators available in C#. Thus a programmer can use operators with user-defined types as well
Others	Manual Garbage Collection. C++ perfomance is the best	Native Garbage Collection. Java performance is slower than C++ due to the native of JIT languages.	Native Garbage Collection. C# is slower than C++ in the same way that Java is slower than C++, but we cant compare C# and Java performance.