**Differentiate the following:**

1. **Class and Object.**
2. **Array and Collection**
3. **Method overriding and method overloading.**
4. **Class and Structure.**
5. **Constructor and Methods.**

**ANSWERS:**

**What is the relationship between a class and an object?**

**A class acts as a blue-print that defines the properties, states, and behaviors that are common to a number of objects.**

**An object is an instance of the class. For example, you have a class called Vehicle and Car is the object of that class. You can create any number of objects for the class named Vehicle, such as Van, Truck, and Auto.  
  
The new operator is used to create an object of a class. When an object of a class is instantiated, the system allocates memory for every data member that is present in the class.**

**What is the difference between arrays and collection?**

**Array:**

1. **You need to specify the size of an array at the time of its declaration. It cannot be resized dynamically.**
2. **The members of an array should be of the same data type.**

**Collection:**

1. **The size of a collection can be adjusted dynamically, as per the user's requirement. It does not have fixed size.**
2. **Collection can have elements of different types.**

**Class:**

1. **A class is a reference type.**
2. **While instantiating a class, CLR allocates memory for its instance in heap.**
3. **Classes support inheritance.**
4. **Variables of a class can be assigned as null.**
5. **Class can contain constructor/destructor.**

**Structure:**

1. **A structure is a value type.**
2. **In structure, memory is allocated on stack.**
3. **Structures do not support inheritance.**
4. **Structure members cannot have null values.**
5. **Structure does not require constructor/destructor and members can be initialiazed automatically.**

**What are similarities between a class and a structure.**

**Structures and classes are the two most important data structures that are used by programmers to build modular programs by using OOP languages, such as Visual Basic .NET, and Visual C#. The following are some of the similarities between a class and a structure:**

* **Access specifiers, such as public, private, and protected, are identically used in structures and classes to restrict the access of their data and methods outside their body.**
* **The access level for class members and struct members, including nested classes and structs, is private by default. Private nested types are not accessible from outside the containing type.**
* **Both can have constructors, methods, properties, fields, constants, enumerations, events, and event handlers.**
* **Both structures and classes can implement interfaces to use multiple-inheritance in code.**
* **Both structures and classes can have constructors with parameter.**
* **Both structures and classes can have delegates and events.**

**How is method overriding different from method overloading?**

**Overriding involves the creation of two or more methods with the same name and same signature in different classes (one of them should be parent class and other should be child).   
  
Overloading is a concept of using a method at different places with same name and different signatures within the same class.**

**What are methods?**

**Methods are the building blocks of a class, in which they are linked together to share and process data to produce the result. In other words, a method is a block of code that contains a series of statements and represents the behavior of a class. While declaring a method you need to specify the access specifier, the return value, the name of the method, and the method parameters. All these combined together is called the signature of the method.**

**Explain the concept of constructor?**

**Constructor is a special method of a class, which is called automatically when the instance of a class is created. It is created with the same name as the class and initializes all class members, whenever you access the class. The main features of a constructor are as follows:**

* **Constructors do not have any return type**
* **Constructors are always public**
* **It is not mandatory to declare a constructor; it is invoked automatically by .NET Framework.**