1. Write the difference between Abstract class and Interfaces.

|  |  |
| --- | --- |
| **Abstract Class** | **Interfaces** |
| 1. Abstract classdoesn't support multiple inheritance**.** | 1. Interface**supports multiple inheritance.** |
| 1. An abstract class can be extended using keyword "extends" | 1. An **interface** can be implemented using keyword "implements". |
| 1. The abstract keyword is used to declare abstract class. | 1. The **interface keyword** is used to declare interface. |
| 1. Multiple Inheritance is not supported | 1. Interface supports Multiple Inheritance. |
| 1. Abstract class can have any type of members like private, public. | 1. Interface can only have public members. |
|  |  |
|  |  |

1. What is method overriding?

Declaring a **method** in sub class which is already present in parent class is known as **method overriding**. **Overriding** is done so that a child class can give its own implementation to a **method** which is already provided by the parent class.

1. What is method overloading?

Two or more **methods** may have the same name if they differ in parameters (different number of parameters, different types of parameters, or both). These **methods** are called **overloaded methods** and this feature is called **method overloading**.

1. Write a code to create 10 letter random string.
2. What is the basic difference between method and constructor?

|  |  |
| --- | --- |
| **Method** | **Constructor** |
| 1. Constructor is used to initialize an object | 1. Method is used to execute certain statements. |
| 1. A constructor cannot be inherited by a subclass. | 1. A method is inherited by a subclass. |
| 1. A constructor can not have any return type. | 1. A method can have a return type. |
|  |  |
|  |  |
|  |  |
|  |  |

1. What is CallbyRef and CallbyValue.

The **call by reference** method of passing arguments to a function copies the address of an argument into the formal parameter. It means the changes made to the parameter affect the passed argument.

The **call by value** method of passing arguments to a function copies the actual **value** of an argument into the formal parameter of the function. In general, it means the code within a function cannot alter the arguments used to **call** the function.

1. How string manage its Memory?