|  |  |
| --- | --- |
| **Abstract Class** | **Interface** |
| 1. It can have both defined and undefined methods | 1. It can have undefined & static defined methods |
| 2. Members are ‘default’ by default. | 2. Members are ‘public’ by default |
| 3. Data members are not public, ‘static’ and ‘final’ by default | 3. Data members are public, ‘static’ and ‘final’ by default |
| 4. It can have both variables and constants. | 4. It can have only constants |
| 5. It can have both default and Parameterized constructors | 5. It has no constructor |
| 6. Methods are not ‘abstract’ by default | 6. Methods are ‘abstract’ by default |
| 7. Concrete methods in abstract class can be ‘final’ | 7. Methods can’t be ‘final’ in interface. |
| 8. A class can inherit a single abstract class | 8. A class can inherit multiple interfaces |
| 9. Abstract class doesn’t support Multiple Inheritance | 9.Supports Multiple Inheritance |
| 10. A class can inherit abstract class using ‘extends’ keyword | 10. A class can inherit interface(s) using ‘implements’ keyword |
| 11. Loosely coupled applications cannot be developed | 11. Loosely coupled applications can be developed |
| 12. It is a General Concept and is available in almost all OOP Languages | 12. They are introduced first in java. Now they are available in later version of OOP languages like C# |
| 13. Abstract method overridden in child class can have same or stronger access level | 13. Interface methods overridden in implementing class must have ‘public’ access level |
| 14. Data members can be public, private, protected or default | 14 Data members can only be ‘public’ (Either Implicitly or Explicitly) |
| 15. A class can only extend another class not interface | 15. An interface can only extend another interface(s) not class |
| 16. A class is not ‘abstract’ by default | 16. An interface is ‘abstract’ by default |
| 17. Abstract class name should be in ‘Noun’ form | 17. Interface name can be in ‘Adjective’ or Noun form |
| 18. Super most class of every abstract class is ‘Object’ class | 18. Interface does not have any built in super most interface |
| 19. Suitable for code reusability and to force method overriding. | 19. Suitable for reusing constants and forcing method overriding. |
| 20. Declared using the keyword ‘abstract class’ | 20. Declared using the keyword ‘interface’ |