|  |  |
| --- | --- |
| **Interface** | **Abstract class** |
| 1. **Interface never allows IMPLEMENTATION**   namespace ConsoleApp3  {  Interface Imakemytrip  {  void print()  {  //illegal  }  }  class Program : Imakemytrip  {  static void Main(string[] args)  {    }  }  } | 1. **ABSTRACT class allows IMPLEMENTATION**   namespace ConsoleApp3  {  abstract class makemytrip  {  public void print()  {  //LEGAL  }  }  class Program : makemytrip  {  static void Main(string[] args)  {    }  }  } |
| 1. **Interface HAS NO MODIFIER**   namespace ConsoleApp3  {  Interface Imakemytrip  {  void print();  }  class Program : Imakemytrip  {  Public void print  {  }  static void Main(string[] args)  {    }  }  } | 1. **ABSTRACT CLASS HAS MODIFIERS.**   namespace ConsoleApp3  {  abstract class makemytrip  {  public void print();  }  class Program : makemytrip  {  static void Main(string[] args)  {    }  }  } |
| 1. **By default INTERFACE MEMBERS ARE, PRIVATE.** | 1. **By default INTERFACE MEMBERS ARE, PUBLIC.** |
| 1. **NO FIELD is allowed**   namespace ConsoleApp3  {  Interface Imakemytrip  {  Int a; //NO FIELD IS ALLOWED  void print();  }  class Program : Imakemytrip  {  Public void print  {  }  static void Main(string[] args)  {    }  }  } | 1. **Can have a field**   namespace ConsoleApp3  {  abstract class makemytrip  {  Int a; //field is allowed  public void print();  }  class Program : makemytrip  {  static void Main(string[] args)  {    }  }  } |
| 1. **An INTERFACE can inherit only from INTERFACE.**   **Not from ABSTRACT CLASS.**  abstract class makemytrip  {  public void print()  {  }  }  interface I1 : makemytrip //ILLIGAL  {  } | 1. **An abstract class can inherit from both ABSTRACT CLASS & INTERFACE.**   abstract class makemytrip :I1, I2  {  public void print()  {  }  }  interface I1  {  } |