1. We cannot achieve run time polymorphism via data members because overriding only applies to methods.
2. We cannot make top level classes private, but creating a class within a class can be made private if we only intend to use it within the outer class it is nested within.
3. Abstraction means only having the most relevant details available and hiding the others on the high level, from a design perspective. Encapsulation, however, is more about the specific access to the details such as variables or methods based on the implementation of a class.
4. Yes, because the method of the superclass is always inherited and if we use a different method signature, then we will overload the superclass method.
5. 1. Compiles and runs printing “Hello”

2.Protected modifier

3.Will Compile and “Base” is printed

4.Local Variables cannot be declared static

5. void method() {} and void method(int i) {}