1. Run time polymorphism by data members is not achievable in java. Run time polymorphism is through methods.
2. Yes, inner classes can be declared as private, but outer classes cannot. If outer classes were private there will be no way of accessing them.
3. Abstraction is letting functionality be used without revealing the internal implementation. Encapsulation is grouping together methods and fields and making them accessible through one container.
4. Two methods are overloaded if they have the same method name but different argument lists. A subclass can overload a superclass method if it inherits that method and defines a method with the same name, but different argument list.
    a. Superclass method: void print()
    b. Subclass method: void print(String s)
5. A method can only override an inherited method. It must have the same method name, arguments, and return type. It cannot have a more restrictive access modifier or throw new or broader checked exceptions.

1. B. Runtime error
2. protected
3. D. Will compile and run printing “Base”
4. D. Local variables cannot be declared as static.
5. A. void method(){} and
   C. void method(int i) {}