1. **Value Type:** A Value Type stores its contents in memory allocated on the stack. When you created a Value Type, a single space in memory is allocated to store the value and that variable directly holds a value.

**Reference Type:** Reference Types are used by a reference which holds a reference (address) to the object but not the object itself. Because reference types represent the address of the variable rather than the data itself, assigning a reference variable to another doesn't copy the data.

1. Variables that are based on value types directly contain values. Assigning one value type variable to another copies the contained value. This differs from the assignment of reference type variables, which copies a reference to the object but not the object itself.
2. Reference parameters don't pass the values of the variables used in the function member invocation - they use the variables themselves. Rather than creating a new storage location for the variable in the function member declaration, the same storage location is used, so the value of the variable in the function member and the value of the reference parameter will always be the same. Reference parameters need the ref modifier as part of both the declaration and the invocation - that means it's always clear when you're passing something by reference.
3. A value type stores its contents in memory allocated on the stack.  A reference type always goes on the Heap.
4. The main purpose of mutation testing approach is to check a test suite quality in terms of the adequacy to killing programs with inserted programming faults. Value types and reference type can be either mutable.
5. The ref keyword causes an argument to be passed by reference, not by value. The effect of passing by reference is that any change to the parameter in the called method is reflected in the calling method. The out keyword causes arguments to be passed by reference. This is like the [ref](https://msdn.microsoft.com/en-us/library/14akc2c7.aspx) keyword, except that ref requires that the variable be initialized before it is passed. To use an outparameter, both the method definition and the calling method must explicitly use the out keyword.