Part 1

1. A person's telephone number: **int**
   A person's height: **float**
   A person's age: **int**
   A person's gender (Male, Female, Prefer Not To Answer): **string**
   A person's salary: **decimal**
   A book's ISBN: **string**
   A book's price: **decimal**
   A book's shipping weight: **float**
   A country's population: **long**
   The number of stars in the universe: **long**
   The number of employees in each of the small or medium businesses in the United Kingdom (up to about 50,000 employees per business): **int**

2. **Value types** are stored in the stack memory, which are passed by values, and they are initialized to their default values.
   Reference types are stored in the heap memory, which are passed by references, and they are initialized to NULL

3. Managed resource is automatically handled by the garbage collector
   Unmanaged resource requires manual management by the developer using the IDisposable interface.

4. The purpose of the Garbage Collector is to automatically manage the memory used by managed objects so that we don't need to manually release memory, which helps prevent memory-related errors such as memory leaks and dangling pointers.

Part 2

1. It will throw an exception "***System.DivideByZeroException***"
2. It returns "∞", the infinity symbol.
3. It depends, for example, if we set an int variable to its max value (2147483647), and we add 1, it returns -2147483648; if we add 100, it returns -2147483549
4. x = y++ assigns the value of y to x before incrementing y, and x = ++y increments y before assigning its value to x.
5. **Break** statement terminates the loop and exits the loop immediately.
   Continue statement skips the current iteration of the loop and jumps to the next iteration.
   Return statement terminates the entire function, including any loops it contains, and returns control to the caller.
6. Initialization, Condition, and Iteration. Initialization and condition parts are required
7. The = operator is used to assign a value to a variable, and the == operator is used to check if two values are equal.
8. Yes, but it is an infinite loop

9. The underscore is used as a "discard" or "catch-all" pattern to match any value that doesn't match any of the other patterns in the switch expression.
10. The IEnumerable interface

```
int max = 500;

for(byte i = 0; i < max; i ++)
{
	WriteLine(i);
}
```

First, there will be an error, that says “WriteLine” does not exist.
If we change it to “Console.WriteLine”, we will have a compile-time error because the i variable is declared as a byte type, which has a range of 0 to 255, and the loop condition i < max will always evaluate to true because max is less than 256.

To warn about the problem without changing the code, I can think of adding a check for when i is equal to zero inside the loop, which would indicate that the overflow has occurred. I have tried “try” and “catch”, and also added the “if” condition, but it still didn't work.