C# Assignment 1 Introduction to C# and Data Types

Understanding Data Types

1. What type would you choose for the following “numbers”?
   - A person’s telephone number string
   - A person’s height decimal
   - A person’s age enum
   - 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 decimal
   - A country’s population uint
   - The number of stars in the universe ulong
   - The number of employees in each of the small or medium businesses in the United Kingdom (up to about 50,000 employees per business) ushort
2. What are the difference between value type and reference type variables? What is boxing and unboxing?
   - value type will directly hold the value, while reference type will hold the memory address or reference for this value
   - value types are stored in stack memory, while reference type will be stored in heap memory
   - value type will not be collected by garbage collector, while reference type will be collected by garbage collector (CLR: Common Language Runtime)
   - value type can be created by Struct or Enum, while reference type can be created by class, interface, delegate or array
   - value type cannot accept null values, while reference type can accept null values
3. What is meant by the terms managed resource and unmanaged resource in .NET
   - Managed resource is reference type data that will managed by garbage collector automatically to allocate and release the memory. Unmanaged resource is value data type that will be discard after use.
4. Whats the purpose of Garbage Collector in .NET?

- The garbage collector manages the allocation and release of memory for an application. For developers working with managed code, this means that you don't have to write code to perform memory management tasks. Automatic memory management can eliminate common problems, such as forgetting to free an object and causing a memory leak or attempting to access memory for an object that's already been freed.

Controlling Flow and Converting Types

1. What happens when you divide an int variable by 0? It will show compile time error.
2. What happens when you divide a double variable by 0? It will output infinity.
3. What happens when you overflow an int variable, that is, set it to a value beyond its range? The loop will run forever.
4. What is the difference between x = y++; and x = ++y;? The former one means that the variable y will be incremented after assigning its value to x. The latter one means that the variable y will be incremented before assigning its value to x.
5. What is the difference between break, continue, and return when used inside a loop statement? The break statement: terminates the closest enclosing iteration statement or switch statement. The continue statement: starts a new iteration of the closest enclosing iteration statement. The return statement: terminates execution of the function in which it appears and returns control to the caller.
6. What are the three parts of a for statement and which of them are required? The *initializer* section that is executed only once, before entering the loop. Typically, you declare and initialize a local loop variable in that section. The declared variable can't be accessed from outside the for statement. The *condition* section that determines if the next iteration in the loop should be executed. If it evaluates to true or is not present, the next iteration is executed; otherwise, the loop is exited. The *condition* section must be a Boolean expression. The iterator section that defines what happens after each execution of the body of the loop. All sections all optional and are not required.
7. What is the difference between the = and == operators? The "=" is an assignment operator is used to assign the value on the right to the variable on the left. The '==' operator checks whether the two given operands are equal or not. If so, it returns true. Otherwise it returns false.
8. Does the following statement compile? for (; true;) ; Yes

9. What does the underscore _ represent in a switch expression? The underscore (_) character replaces the **default** keyword to signify that it should match anything if reached.
10. What interface must an object implement to be enumerated over by using the foreach statement? IEnumerable interface