Understanding Data Types

Test your Knowledge

1. What type would you choose for the following “numbers”?
   short

2. A person’s telephone number
   string
3. A person’s height
   decimal
4. A person’s age
   byte
5. A person’s gender (Male, Female, Prefer Not To Answer)
   enum
6. A person’s salary
   decimal
7. A book’s ISBN
   string
8. A book’s price
   decimal
9. A book’s shipping weight
   decimal
10. A country’s population
    long
11. The number of stars in the universe
    BigInteger
12. 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. What are the differences between value type and reference type variables? What is

boxing and unboxing?

The difference between value types and reference types are as follows:

1. Value types are the datatypes which stores the value whereas reference types are the datatypes which store the reference of values,
2. Value types store data in stack memory whereas reference types store values in the heap memory(the reference is stored in stack).
3. Value types cannot contain null values whereas reference types can contain null values.
4. Examples of Value types are int, long, short,char, enum etc whereas reference types are string, classes, interfaces etc.
5. Value types have default values like int has 0, boolean has false whereas the default value of reference type is null.
6. Value type are more performant for simple and small data due to stack allocation and memory management whereas reference types are useful for complex data structures.
7. Value types are automatically removed from stack once unused whereas reference types are removed by the garbage collector once no longer referenced.

Boxing and Unboxing:
Boxing is the way to convert the value type to it's corresponding reference type when stored in an object whereas unboxing is the way to convert the reference type to its corresponding value type.
Boxing allows value types to be treated as objects. It is suitable in scenarios where we need to store value types in reference types. In the other had unboxing is suitable in the scenarios where we no longer need reference of the object but its original value.
Boxing is done automatically meaning it doesn't need casting but Unboxing is done implicitly meaning it requires casting to correct value type.

3. What is meant by the terms managed resource and unmanaged resource in .NET
Managed resource in .Net
In .NET, managed resources are ones that .NET runtime and garbage collector (GC) handle by itself, like strings, arrays, or lists. These get allocated and cleaned up automatically, so developer no need to do much. But unmanaged resources are not handled by GC, like file handles, database connections, or network sockets. These need manual cleanup to avoid leaks. To manage unmanaged resources, developers use IDisposable and write a Dispose method to clean them up properly.

4. Whats the purpose of Garbage Collector in .NET?
The purpose of Garbage Collector in .Net is to automatically manage memory. It collect the unused reference types and remove them once they are no longer in use. It helps to prevent memory leaks and improve the performance.

Controlling Flow and Converting Types

Test your Knowledge

1. What happens when you divide an int variable by 0?
   I get DivideByZeroException

2. What happens when you divide a double variable by 0?
   It depends on the sign of the numerator. If the denominator is positive i ger PositiveInfinity whereas if the denominator is negative I get NegativeInfinity.
   If the denominator is 0 then I get NaN, Not a Number

3. What happens when you overflow an int variable, that is, set it to a value beyond its range?
   I get StackoverFlowExcerion

4. What is the difference between x = y++; and x = ++y;?
   The difference is the x = y++ adds 1 after y whereas x=++y adds 1 before y.

5. What is the difference between break, continue, and return when used inside a loop Statement?
   Break ends the loop and moves forward to next statement. Continue skips the current iteration moves to next iiteration and reruns the loop and return exists the function.

6. What are the three parts of a for statement and which of them are required?
   The three parts are
   Declaration of starting point,(Initialization)
   Increment or decrement,(Update)
   End of the point at which the increment or decrement should reach (Condition)
   We dont need update and initilization as required.

7. What is the difference between the = and == operators?
   = is assignment operator whereas == checks if the references or values match

8. Does the following statement compile? for (; true;) ;
   Yes.

9. What does the underscore _ represent in a switch expression?
   It is substitute for default

10. What interface must an object implement to be enumerated over by using the foreach statement?
    IEnumerable