Understanding Data Types

1.

Ans:

* A person’s telephone number: String.
* A person's height: Decimal.
* A person's age: int.
* A person's gender (Male, Female, Prefer Not to Answer):
* 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: Float.
* The number of stars in the universe: Double.
* The number of employees in each of the small or medium businesses in the United Kingdom (up to about 5,000 employees per business): Int.

2. What are the differences between value type and reference type variables? What is boxing and unboxing?

Ans:

Variables of reference types store references to their data (objects), while variables of value types directly contain their data. With reference types, two variables can reference the same object; therefore, operations on one variable can affect the object referenced by the other variable.

Boxing is the process of converting a value type to the type object or to any interface type implemented by this value type. When the common language runtime (CLR) boxes a value type, it wraps the value inside a System. Object instance and stores it on the managed heap. Unboxing extracts the value type from the object.

3. What is meant by the terms managed resource and unmanaged resource in .NET

Ans:

Managed resources are those that are pure .NET code and managed by the runtime and are under its direct control.

Unmanaged resources are those that are not .NET File handles, pinned memory, COM objects, database connection.

4. What is the purpose of Garbage Collector in .NET?

Ans:

The purpose of Garbage Collector in .NET is to manage the allocation and release of memory for application.

Controlling Flow and Converting Types

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

Ans: It will throw a DivideByZeroException exception.

2. What happens when you divide a double variable by 0?

Ans:

It won’t throw a result.

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

Ans:

An integer overflow can cause the value to wrap and become negative.

4. What is the difference between x = y++; and x = ++y?

Ans:

x = y++ happens after assignment. x = ++y happens before assignment.

5. What is the difference between break, continue, and return when used inside a loop statement?

Ans:

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?

Ans:

1. the keyword For that starts the loop
2. the condition being tested
3. the EndFor keyword that terminates the loop

7. What is the difference between the = and == operators?

Ans:

“=” operator is used to assign value to a variable .

“==” operator is used to compare two variable or constants.

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

Ans:

The statement for (; true;) will check if true is true creating an infinite loop.

9. What does the underscore \_ represent in a switch expression? Defaults

Ans:

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?

Ans:

IEnumerable interface.

Arrays and Strings

Test your knowledge

1. When to use String vs. StringBuilder in C# ?

Ans:

If the string we store doesn't need to change, use String;

If the string we store needs to be changeable, use StringBuilder

2. What is the base class for all arrays in C#?

Ans:

Array Class.

3. How do you sort an array in C#?

Ans:

Array.Sort().

4. What property of an array object can be used to get the total number of elements in an array?

Ans:

The Length property.

5. Can you store multiple data types in System.Array?

Ans:

object[] array = new object[];

6. What’s the difference between the System.Array.CopyTo() and System.Array.Clone()?

Ans:

The Clone() method returns a new array (a shallow copy) object containing all the elements in the original array.

The CopyTo() method copies the elements into another existing array. Both perform a shallow copy.