1. Calculate the number of bytes a structure variable "s" occupies in the memory (on 64-bit system) if it is defined as follows.

```
class trial {
    int i;
    Decimal d;
}
struct sample {
    private int x;
    private Single y;
    private trial z;
}
// in Main()
sample s = new sample();
```

- A. 24 bytes
- B. 8 bytes
- C. 16 bytes
- D. 12 bytes
- Answer: C

---

2. Which among the following cannot be used as a datatype for an enum in C#.NET?
    1. short
    2. double
    3. byte
    4. Object
    5. bool
    6. int

- A. 1,2,3,5
- B. 2,4,5
- C. 2,3,4,5
- D. 1,2,4,6
- Answer: B

Q3.Two namespaces n1 and n2 each contain a class named Demo. Which option correctly creates separate objects x and y from both classes:-

- A.

```
using n1;
using n2;

Demo x = new n1.Demo();
Demo y = new n2.Demo();
```

- B.

```
using n1;
using n2;

n1.Demo x = new Demo();
n2.Demo y = new Demo();
```

- C.

```
using Demo1 = n1.Demo;
using Demo2 = n2.Demo;

Demo1 x = new Demo1();
Demo2 y = new Demo2();
```

- D.

```
import n1;
import n2;

Demo x = new n1.Demo();
Demo y = new n2.Demo();
```

- Answer: C

---

4. Which among the following is a correct statement about namespace used in C#.NET?

- A. Classes must belong to a namespace, whereas structures need not.
- B. All elements of the namespace must belong to one file
- C. By default, namespace name is same a project name in Visual Studio.
- D. All of the mentioned
- Answer: C

---

5. Which class Greet() method will get called?

```
class Root {
    public new virtual void Greet() { }
}

class Middle : Root {
    public override void Greet() { }
}

class Leaf : Middle
{
}

class Program
{
    static void Main() {
        Root obj = new Leaf();
        obj.Greet();
    }
}
```

- A. Leaf
- B. Middle
- C. Root
- D. Compile-Time Error
- Answer: B

---

6. Which of the following statements about ref, out, and in parameters in .NET is incorrect?

- A. ref parameter provides two way communication between the caller and method.
- B. In ref as well as out parameter, method can return a value.

- C. An out argument must be assigned in the called method before returning, while a ref argument must be assigned in the calling method before passing.
- D. The ref and out keywords are interchangeable, as both allow bidirectional data flow between methods.

Answer: D

---

7. What will happen when OuterMethod() is called?

```
void OuterMethod() {
    int x = 10;

    static void InnerMethod() {
        Console.WriteLine(x);
    }

    InnerMethod();
}
```

- A. The code fails to compile because static local functions cannot capture local variables from the enclosing method.
- B. The code compiles and prints 10, because static local functions capture outer variables like lambdas.
- C. The code compiles, but throws a runtime error since static methods delay variable resolution until runtime.
- D. The code compiles and runs successfully by automatically promoting x to a static field in the enclosing class.
- Answer: A

---

8. A company, "Acme Corp," has multiple teams working on different parts of their e-commerce platform. Each team has developed classes, structs, and other code related to their specific area. Team A is responsible for product catalog management, Team B handles user accounts, and Team C manages shopping carts. How should the company best organize their code to avoid naming conflicts and ensure maintainability, particularly when these teams might eventually need to interact with each other's code?

- A. Place all classes, structs, and other code into a single global namespace called "AcmeCorp".
- B. Create separate namespaces for each team, such as "AcmeCorp.ProductCatalog", "AcmeCorp.UserAccounts", and "AcmeCorp.ShoppingCart".
- C. Use different access modifiers (private, public, etc.) to manage visibility between the teams' code.
- D. Place all code into a single namespace and rely on unique class names within that namespace.
- Answer: B

---

9. Choose the correct statement.

- A.The code inside #region will only compile Debug mode; it is skipped entirely in Release mode.
- B. #region and #endregion define code that can be collapsed in the Visual Studio editor, but have no effect on compilation or runtime behavior.
- C. #region automatically wraps the code inside a try-catch block to simplify debugging in the editor.
- D. Using #region disables syntax errors inside the block until #endregion is reached, allowing partial compilation.
- Answer: B

---

10. Why we use properties instead of simple getter and setter in c#?

- A. Properties are required when defining private fields and are only usable within the same class unless marked public
- B. Properties improve performance by allowing the compiler to optimize field access at runtime and bypass method calls entirely.
- C. Properties provide a familiar field-like syntax while offering access restrictions and input verification.
- D. Properties enforce compile-time type safety in a way that getter and setter methods cannot, due to CLR-level enforcement rules.
- Answer: C
