**1. What is the primary purpose of reflection in C#?**

a) To modify the source code at runtime  
b) To obtain type information at runtime  
c) To improve the performance of a program  
d) To handle exceptions more effectively  
**Answer:** b) To obtain type information at runtime

**2. Which namespace contains the reflection-related classes in C#?**

a) System.Type  
b) System.Reflection  
c) System.Dynamic  
d) System.Diagnostics  
**Answer:** b) System.Reflection

**3. What can reflection be used for in C#?**

a) Viewing attribute information at runtime  
b) Instantiating types dynamically  
c) Late binding to methods and properties  
d) All of the above  
**Answer:** d) All of the above

**4. Which class in the System.Reflection namespace is used to get metadata about a type?**

a) TypeInfo  
b) MethodInfo  
c) FieldInfo  
d) PropertyInfo  
**Answer:** a) TypeInfo

**5. How do you obtain the type information of a class using reflection?**

a) Type t = new Type(MyClass);  
b) Type t = typeof(MyClass);  
c) Type t = MyClass.GetType();  
d) Type t = Reflection.GetType(MyClass);  
**Answer:** b) Type t = typeof(MyClass);

**6. What does the GetCustomAttributes method do?**

a) Retrieves metadata about a class  
b) Returns an array of attributes applied to a class  
c) Modifies the attributes of a class at runtime  
d) Deletes the attributes of a class  
**Answer:** b) Returns an array of attributes applied to a class

**7. Which reflection class is used to get method information?**

a) TypeInfo  
b) MethodBase  
c) MethodInfo  
d) PropertyInfo  
**Answer:** c) MethodInfo

**8. What is required to retrieve attribute information using reflection?**

a) The MemberInfo object  
b) The AttributeInfo object  
c) The ReflectionInfo object  
d) The TypeMetadata object  
**Answer:** a) The MemberInfo object

**9. How can you retrieve all methods of a class using reflection?**

a) Type.GetMethods()  
b) Type.FindMethods()  
c) Type.GetAllMethods()  
d) Type.Methods()  
**Answer:** a) Type.GetMethods()

**10. What is the output of the following reflection code snippet?**

Type t = typeof(MyClass);

object[] attributes = t.GetCustomAttributes(true);

foreach (var attr in attributes)

{

Console.WriteLine(attr);

}

a) It prints the names of the attributes applied to MyClass  
b) It prints the methods of MyClass  
c) It throws a runtime error  
d) It prints the fields of MyClass  
**Answer:** a) It prints the names of the attributes applied to MyClass