### Reflection

1. \*\*What does Reflection in C# allow you to do?\*\*

- A) Access private fields

- B) Access metadata of assemblies

- C) Execute SQL queries

- D) Generate random numbers

\*\*Answer: B) Access metadata of assemblies\*\*

2. \*\*Which class is used to obtain information about a type at runtime?\*\*

- A) Type

- B) TypeInfo

- C) Assembly

- D) MemberInfo

\*\*Answer: A) Type\*\*

3. \*\*Which method of the `Type` class is used to get the full name of a type?\*\*

- A) GetType()

- B) FullName()

- C) ToString()

- D) Name()

\*\*Answer: B) FullName()\*\*

4. \*\*How can you get a list of all methods in a type using Reflection?\*\*

- A) `type.GetFields()`

- B) `type.GetProperties()`

- C) `type.GetMethods()`

- D) `type.GetConstructors()`

\*\*Answer: C) type.GetMethods()\*\*

5. \*\*Which class is used to invoke methods on a type at runtime?\*\*

- A) MethodInfo

- B) FieldInfo

- C) PropertyInfo

- D) ConstructorInfo

\*\*Answer: A) MethodInfo\*\*

6. \*\*What does `Activator.CreateInstance()` do?\*\*

- A) Creates a new instance of a type

- B) Retrieves the assembly of a type

- C) Loads a type into memory

- D) Gets metadata of a type

\*\*Answer: A) Creates a new instance of a type\*\*

7. \*\*Which attribute can be used to define metadata about a class?\*\*

- A) [Metadata]

- B) [Serializable]

- C) [Description]

- D) [DataContract]

\*\*Answer: B) [Serializable]\*\*

8. \*\*Which class provides methods to get information about the members of a type?\*\*

- A) Type

- B) MemberInfo

- C) Assembly

- D) Attribute

\*\*Answer: B) MemberInfo\*\*

9. \*\*To obtain information about fields of a type, which method is used?\*\*

- A) `GetMethods()`

- B) `GetFields()`

- C) `GetProperties()`

- D) `GetConstructors()`

\*\*Answer: B) GetFields()\*\*

10. \*\*Which interface is implemented by classes that provide information about a type’s members?\*\*

- A) ITypeInfo

- B) IReflect

- C) IMetadata

- D) IAssembly

\*\*Answer: B) IReflect\*\*

### Attributes

11. \*\*What is an attribute in C#?\*\*

- A) A type of variable

- B) Metadata attached to code elements

- C) A method of a class

- D) A special type of exception

\*\*Answer: B) Metadata attached to code elements\*\*

12. \*\*How do you apply an attribute to a class in C#?\*\*

- A) [Attribute]

- B) <Attribute>

- C) (Attribute)

- D) @Attribute

\*\*Answer: A) [Attribute]\*\*

13. \*\*Which attribute is used to indicate that a class can be serialized?\*\*

- A) [DataContract]

- B) [Serializable]

- C) [JsonObject]

- D) [XmlRoot]

\*\*Answer: B) [Serializable]\*\*

14. \*\*How can custom attributes be retrieved at runtime?\*\*

- A) Using Reflection

- B) Using a specific compiler directive

- C) By examining the code directly

- D) By parsing XML files

\*\*Answer: A) Using Reflection\*\*

15. \*\*Which attribute is used to define a custom attribute?\*\*

- A) [CustomAttribute]

- B) [AttributeUsage]

- C) [Attribute]

- D) [Metadata]

\*\*Answer: C) [Attribute]\*\*

16. \*\*Which property of the `Attribute` class provides information about the attribute’s usage?\*\*

- A) Usage

- B) AttributeTargets

- C) Target

- D) Metadata

\*\*Answer: B) AttributeTargets\*\*

17. \*\*To create a custom attribute, which base class should be derived from?\*\*

- A) System.Attribute

- B) System.Object

- C) System.ComponentModel

- D) System.MarshalByRefObject

\*\*Answer: A) System.Attribute\*\*

18. \*\*Which attribute specifies where a custom attribute can be applied?\*\*

- A) [CustomUsage]

- B) [Location]

- C) [AttributeUsage]

- D) [Scope]

\*\*Answer: C) [AttributeUsage]\*\*

19. \*\*What does the `AllowMultiple` property in the `AttributeUsage` attribute specify?\*\*

- A) Whether the attribute can be applied to multiple members

- B) Whether the attribute can be inherited

- C) Whether the attribute can be used on both fields and properties

- D) Whether the attribute can be overridden

\*\*Answer: A) Whether the attribute can be applied to multiple members\*\*

20. \*\*Which attribute is used to mark methods that are obsolete?\*\*

- A) [Obsolete]

- B) [Deprecated]

- C) [Outdated]

- D) [DeprecatedMethod]

\*\*Answer: A) [Obsolete]\*\*