1. Access modifier:

public: Members declared as public are accessible from any other class.

protected: Members declared as protected are accessible within the same package or by subclasses in different packages.

Default (no modifier): If no access modifier is specified, the member is accessible only within the same package (package-private).

private: Members declared as private are accessible only within the same class.

These modifiers determine the accessibility of classes, methods, and variables in Java:

Class Access: The access modifiers can be applied to classes. Only the public and default access modifiers can be applied to classes. Public classes are accessible from anywhere, while default classes are accessible only within the same package.

Method Access: Access modifiers can be applied to methods. All four access modifiers can be used for methods. Public methods are accessible from anywhere, protected methods are accessible within the same package and by subclasses, default methods are accessible only within the same package, and private methods are accessible only within the same class.

Variable Access: Access modifiers can be applied to variables. Again, all four access modifiers can be used. Public variables are accessible from anywhere, protected variables are accessible within the same package and by subclasses, default variables are accessible only within the same package, and private variables are accessible only within the same class.

2.Difference Between

Exceptions: Exceptions represent exceptional conditions that can occur during the execution of a program. These conditions are usually caused by user input errors, invalid operations, or unexpected situations.

Exceptions are recoverable, meaning that the program can catch and handle them gracefully. They can be caught using try-catch blocks, allowing the program to continue execution or take appropriate action.

Errors: Errors represent serious, usually unrecoverable problems that occur at runtime and are typically beyond the control of the application. They often indicate problems that the application cannot handle or should not attempt to recover from.

Errors are not meant to be caught or handled by the application code.

3.Difference Between

Checked Exceptions: Checked exceptions are the exceptions that are checked by the compiler at compile time to ensure that they are caught or declared to be thrown by the method.

These exceptions are typically caused by external factors outside the control of the program, such as I/O errors, network issues, and file handling errors.

Unchecked Exceptions: Unchecked exceptions are exceptions that do not need to be explicitly handled by the programmer at compile time. The compiler does not enforce catching or declaring these exceptions.

Unchecked exceptions are exceptions that do not need to be explicitly handled by the programmer at compile time. The compiler does not enforce catching or declaring these exceptions.