Task 11

1. Access modifiers in java

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.

1. Diff. between the Exception and Error

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.

Exceptions are subclasses of the Exception class, which itself extends the Throwable class.

Examples of exceptions include NullPointerException, ArrayIndexOutOfBoundsException, FileNotFoundException, etc.

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. They typically indicate issues at the system level or severe problems with the runtime environment.

Errors are subclasses of the Error class, which also extends the Throwable class.

Examples of errors include OutOfMemoryError, StackOverflowError, VirtualMachineError, etc.

1. Diff. btw checked and unchecked exceptions

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.

Checked exceptions are subclasses of Exception but not subclasses of RuntimeException.

Examples of checked exceptions include IOException, FileNotFoundException, SQLException, etc.

When a method can potentially throw a checked exception, the caller method must either catch the exception using a try-catch block or declare that it throws the exception using the throws clause in its method signature.

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 usually represent programming errors or logic errors within the program, such as null pointer dereferences, invalid array indexes, and arithmetic exceptions.

Unchecked exceptions are subclasses of RuntimeException or its subclasses.

Examples of unchecked exceptions include NullPointerException, ArrayIndexOutOfBoundsException, ArithmeticException, etc.

While it's not mandatory to catch or declare unchecked exceptions, it's still good practice to handle them gracefully to prevent unexpected program termination