Program: Demonstrating `throw` and `throws` in Java

```java
class ExceptionExample {

    // Method that declares it may throw an exception
    public void checkAge(int age) throws IllegalArgumentException {
        if (age < 18) {
            // Using throw to manually throw an exception
            throw new IllegalArgumentException("Age must be 18 or older.");
        } else {
            System.out.println("Access granted.");
        }
    }

    // Method that handles an exception using try-catch block
    public void divide(int a, int b) throws ArithmeticException {
        if (b == 0) {
            // Throwing an ArithmeticException if denominator is zero
            throw new ArithmeticException("Cannot divide by zero.");
        }
        System.out.println("Result: " + (a / b));
    }
}

public class Main {
    public static void main(String[] args) {
        ExceptionExample example = new ExceptionExample();

        // Handling exception using try-catch block for checkAge method
        try {
            example.checkAge(16); // This will throw an IllegalArgumentException
        } catch (IllegalArgumentException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }

        // Handling exception using try-catch block for divide method
        try {
            example.divide(10, 0); // This will throw an ArithmeticException
        } catch (ArithmeticException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
        // Successful execution without exceptions
        try {
            example.checkAge(20); // This will pass
            example.divide(10, 2); // This will pass
        } catch (Exception e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}
```

Explanation

- `throws` **keyword**:
    - The `checkAge` method declares that it may throw an `IllegalArgumentException` by using the `throws` keyword. This means that any code calling `checkAge` must handle this exception (either by catching it or declaring it further up the chain).
    - Similarly, `divide` method declares that it may throw an `ArithmeticException` using `throws`.
- `throw` **keyword**:
    - Inside `checkAge`, we use the `throw` keyword to manually throw an `IllegalArgumentException` if the `age` is less than 18.
    - In the `divide` method, `throw` is used to manually throw an `ArithmeticException` if the divisor (`b`) is zero.
- **try-catch blocks**:
    - In `main`, we call `checkAge` and `divide` inside `try` blocks to handle potential exceptions.
    - If an exception is thrown, it is caught by the corresponding `catch` block, and an error message is printed.

Output

The output for this program would be:

```
Exception caught: Age must be 18 or older.
Exception caught: Cannot divide by zero.
Access granted.
Result: 5
```

Key Points

1. **throws** is used in the method declaration to specify that this method might throw exceptions of a specified type.
2. **throw** is used within a method to actually throw an exception, often as part of custom error handling.
3. **try-catch blocks** handle exceptions that might be thrown by methods, allowing the program to continue running gracefully.

This example shows how `throws` and `throw` work together to manage error conditions in Java.