1. Create an ArrayList<Object> and add the following elements:

- Integer value 42

- Double value 3.14

- String "1000"

- Fraction(9, 4)

Write a method that:

- Iterates through the list

- Tries to convert each element to a numeric double value using appropriate casting and parsing

- Skips non-numeric types with a warning

- Returns the sum as a double

2. Compare the following Integer assignments:

Integer a = 128;

Integer b = 128;

Integer c = 127;

Integer d = 127;

- Print results of `(a == b)` and `(c == d)`

- Explain the behavior based on Java’s Integer caching mechanism.

3. Create a method that accepts two `Integer` objects and compares them using both `==` and `.equals()`.

- Demonstrate the result when values are within cache range (e.g., 100) and outside cache range (e.g., 1000)

4. Use a StringTokenizer with the string: `"12-abc|56,xyz 90"` and delimiters: `"-|, "`

- Extract all numeric tokens

- Convert them to integers

- Return the total sum

5. Write a method that:

- Accepts a string of math expressions like `"3+5-2+8"`

- Tokenizes it using "+-" as delimiters (with flag `true`)

- Separates numbers from operators

- Evaluates the final result (e.g., for "3+5-2+8" it returns 14)

6. Add the following to an ArrayList<Object>: `"123" (String)`, `123 (Integer)`, `"3.14" (String)`, `3.14 (Double)`

- Loop through the list

- Parse all numeric values as double (whether from wrapper or from string)

- Return the average

7. Given a string `"value1=100;value2=200;value3=abc;value4=300"`, tokenize and parse the values.

- Ignore non-numeric entries

- Return total of valid values and count of invalid ones