CSE231 Advanced Computer Programming
Lab 06

1) Write a Java console application that accepts two integers from the user, then print their sum, difference, product, quotient, and division remainder. The application should use exception handling to verify that the numbers are actually integers. In case the provided inputs are invalid, the program should allow the user to retry until he enters integers. Also it should use exception handling to support the case were the second integer (the one used as the divisor) is zero. In this particular case, the application should print "0" for the remainder and "+infinity", "-infinity", or "undefined quantity" for the quotient depending on the value of the first integer (the one used as the dividend).

2) Write a Java console application that accepts two **double values** from the user, then print their sum, difference, product, and quotient. The application should use exception handling to verify that the numbers are actually double values. In case the provided inputs are invalid, the program should allow the user to retry until he enters double values. A zero divisor would not cause any problems in this case as infinity and undefined quantity (NaN) values are already supported by the double type.

3) Write a Java class representing a bank account. The class should encapsulate the following fields: an account number (encoded as a `String`), an account holder name (`String`), a balance (`float`), and a flag indicating whether the account is closed or open (`boolean`). The class should have a constructor initializing all the fields. Note that the flag should not be passed as an argument to the constructor; instead, it should be automatically initialized to indicate that the account is not closed. In addition, the class should have the following methods. Note that if a method cannot perform the required action because the necessary condition is not satisfied, it should to throw a user-defined **unchecked** exception and keep the bank account object(s) unchanged.

    a. `print()`: a public method printing all the fields of a bank account.
    b. `getBalance()`: a public method returning the account balance.
    c. `close()`: a public method marking the account as closed only if the balance is zero and if the account is not already closed.
    d. `reopen()`: a public method marking the account as open only if the account is closed.
    e. `deposit(float amount)`: a public method increasing the balance of the account by a given amount. The amount must be positive and the account must be open.
    f. `withdraw(float amount)`: a public method decreasing the balance of the account by a given amount. The amount must be a positive value less than or equal to the account's balance and the account must be open.

g. `transferTo(Account a, float amount)`: a public method transferring the given positive amount from the current account to the passed account provided that both accounts are open and that the transferring account has a sufficient balance.

h. `printAll(Account[] as)`: a public static method that prints all the accounts of the passed array.

i. `find(Account[] as, String accountNum)`: a public static method returning the index of the account that has the passed account number in an array of accounts. It should return -1 in case a matching account is not found.

Write a Java program that calls these methods under conditions causing them to throw these exceptions without handling them. What will be the output of the program you wrote?

The program will not display any outputs, instead it will throw an InvalidBankAccountStateException with a message "Cannot close an account that has money" thus ending the program.

4) Modify the methods of the bank account class you defined in Exercise 3, to throw user-defined **checked** exceptions instead of displaying error messages when the operation cannot be performed. Write a Java program that calls these methods under conditions causing them to throw these exceptions. What will be the output of the program you wrote?

5) Consider the following program:

```
import java.util.*;


public class Q5 {
    public static void main(String[] args)
{         Scanner sc = new
Scanner(System.in);          int a = 40;
int b = sc.nextInt();          int x = a/b;
        System.out.println("x="+x);
try {
            int c = sc.nextInt();
int d = sc.nextInt();
if(c >= 100){
                int[] nums = new
int[2];                 nums[0]=c;
nums[1]=d;                  nums[2]=c-d;
            }
            int y = c/d;
            System.out.println("y="+c/d);
        }
        catch(InputMismatchException ex){
            System.out.println("Non-integer inputs.");
        }
        catch(ArithmeticException ex){
            System.out.println("Cannot divide by Zero.");
        }
        finally {
            System.out.println("Finally.");
}
        System.out.println("Done.");
}
}
```

What will be the output for each the following inputs?

a. 16, 8, 10
b. "X", 10, 0
c. 0, "X", "Y"
d. 10, "X", 17
e. 7, 150, 0
f. 87, 18, 0