

Tracing BankAccount

Name: _____

Date: _____

Use the BankAccount class from homework. The class below is in a separate file, Bank.java, in the same folder.

```
public class Bank
{
    /** Moves all of one account's money into another. */
    public static void merge(BankAccount from, BankAccount to)
    {
        double amount = from.getBalance();
        from.withdraw(amount);
        to.deposit(amount);
    }

    public static void main(String[] args)
    {
        // lines 1-35 from the table below go here
    }
}
```

For each line, write one of the following in the right-hand column:

- the exact output, if the line prints something;
- **CE** (compile error) and a short reason. Then assume the line was deleted and keep going;
- **RE** (runtime error) and the name of the exception. The program stops there;
- a dash (–) if the line compiles and runs but prints nothing.

Tip: keep track of each account's balance in the margin as you go.

#	Code (inside main in Bank.java)	Output, CE, or RE
1	BankAccount a = new BankAccount("Ada", 200.0);	
2	BankAccount b = new BankAccount("Ben");	
3	BankAccount c = new BankAccount("Cy", 50.0);	
4	BankAccount d = new BankAccount();	
5	System.out.println(BankAccount.getNumAccounts());	
6	System.out.println(c);	
7	b.deposit(75.0);	
8	b.deposit(-30.0);	
9	System.out.println(b.getBalance());	
10	System.out.println(a.balance);	
11	System.out.println(a.withdraw(250.0));	
12	System.out.println(a.transferTo(c, 120.0));	
13	System.out.println(a + " / " + c);	
14	c.setBalance(1000.0);	
15	merge(c, b);	
16	System.out.println(b.getBalance() + c.getBalance());	
17	BankAccount e = b;	
18	e.withdraw(45.0);	

#	Code (inside main in Bank.java)	Output, CE, or RE
19	System.out.println(b.getBalance());	
20	System.out.println(e == b);	
21	System.out.println(a.equals(b));	
22	BankAccount.MIN_BALANCE = -100.0;	
23	BankAccount.setInterestRate(0.5);	
24	b.addInterest();	
25	System.out.println(b);	
26	System.out.println(BankAccount.getBalance());	
27	a.computeInterest();	
28	a.setOwner("Ava");	
29	System.out.println(a);	
30	System.out.println(b.getOwner() + " has \$" + b.getBalance());	
31	BankAccount g = new BankAccount("Gus", 10.0);	
32	System.out.println(g.getAccountNumber());	
33	BankAccount h = null;	
34	h.deposit(5.0);	
35	System.out.println("done");	

Part B: Short answer

- Line 4 doesn't compile. Give two different ways to fix it: one that changes only line 4, and one that changes only BankAccount.
- Line 26 doesn't compile. Rewrite it so it compiles and prints Ada's balance.
- Line 18 calls a method on e, yet line 19 shows that b changed. Explain why.
- Line 32 prints 4, not 5. Why?
- Line 23 changes the interest rate. Which accounts does it affect? Why is it called on BankAccount and not on one of the accounts?
- What would merge(a, a) do to Ada's balance? Trace it.