**Day 25\_Java Assignment**

**1. Problem Description:**

Synchronized Java code

**2. My Solution:**

**public** **class** Account {

**private** **int** id;

**private** **double** balance;

**private** String name;

**public** Account(**int** id, **double** balance, String name) {

**this**.id = id;

**this**.balance = balance;

**this**.name = name;

}

**public** **int** getId() {

**return** id;

}

**public** **double** getBalance() {

**return** balance;

}

**public** String getName() {

**return** name;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **double** transferMoney(Account toAccount, **double** money) {

**if** (**this**.balance < money) {

System.***out***.println("Insufficient money");

} **else** {

**this**.balance -= money;

toAccount.setBalance(toAccount.getBalance() + money);

}

**return** **this**.balance;

}

}

**public** **class** AccountMain {

**public** **static** **void** main(String[] args) {

Account mahaAccount = **new** Account(1, 100000, "Maha");

Account meenuAccount = **new** Account(2, 150000, "Meenu");

System.***out***.println(mahaAccount.getBalance());

System.***out***.println(meenuAccount.getBalance());

mahaAccount.transferMoney(meenuAccount, 5000);

meenuAccount.transferMoney(mahaAccount, 10000);

System.***out***.println(mahaAccount.getBalance());

System.***out***.println(meenuAccount.getBalance());

}

}

**Output:**

100000.0

150000.0

105000.0

145000.0