JAVA LAB FAT Prashanth.S(19MID0020)

Question

Time left 0:30:49

Question **1**

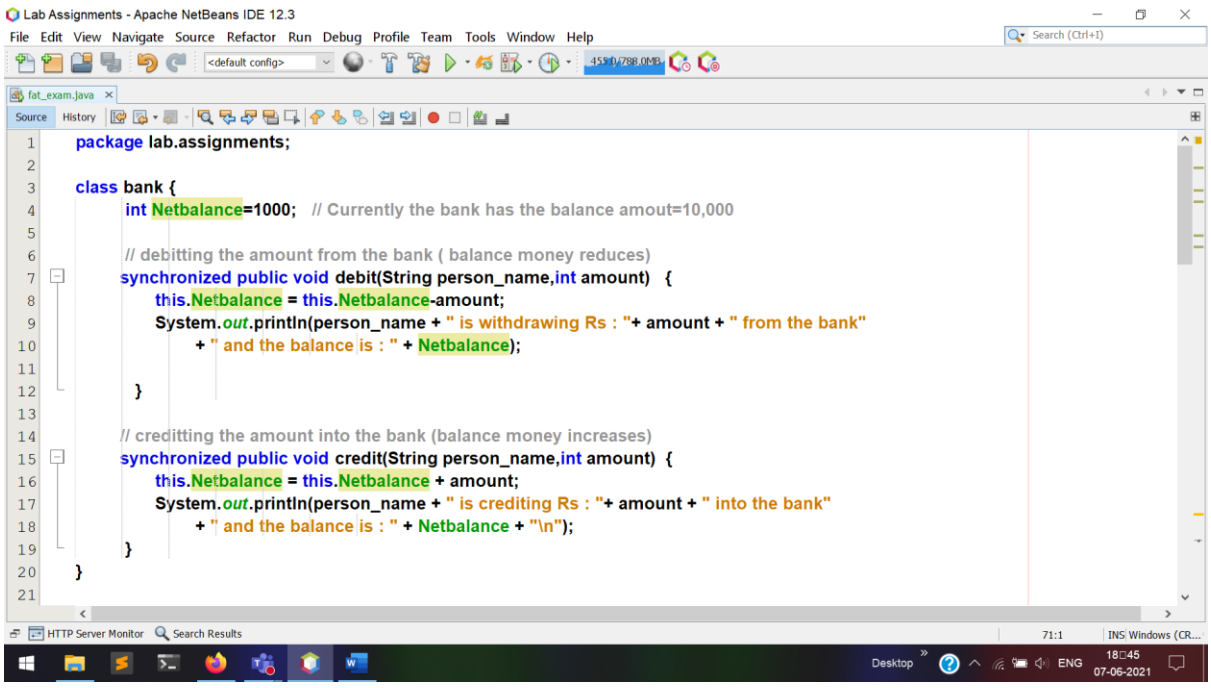
Not yet answered

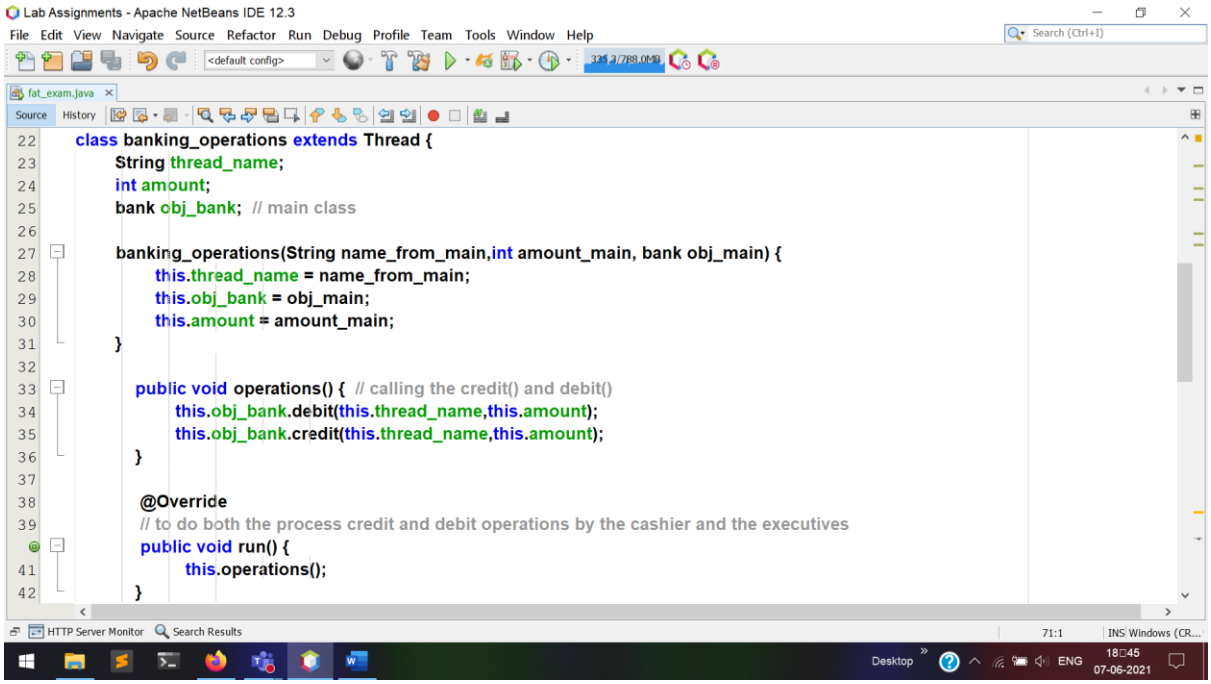
Marked out of 1.00

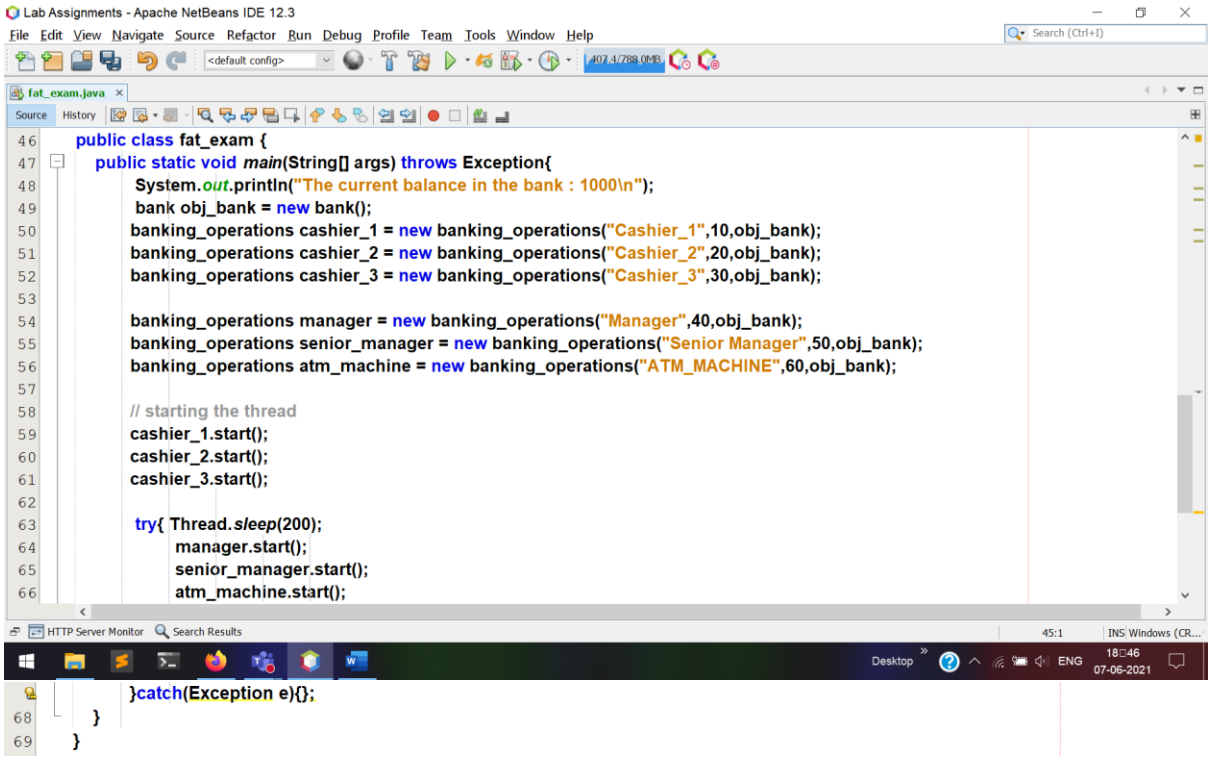
Flag question

Design a java application to implement the banking process. Consider the shared variable NETBALANCE has the complete branch balance. Design the cashier1, cashier2, cashier3 threads to access this variable and allow the operations of credit and debit. Design a manager, senior manager and ATM machine threads, which also has the access to the shared variable and do the operations of credit and debit. Ensure the shared variable is used safely and no multiple access by various threads at time.

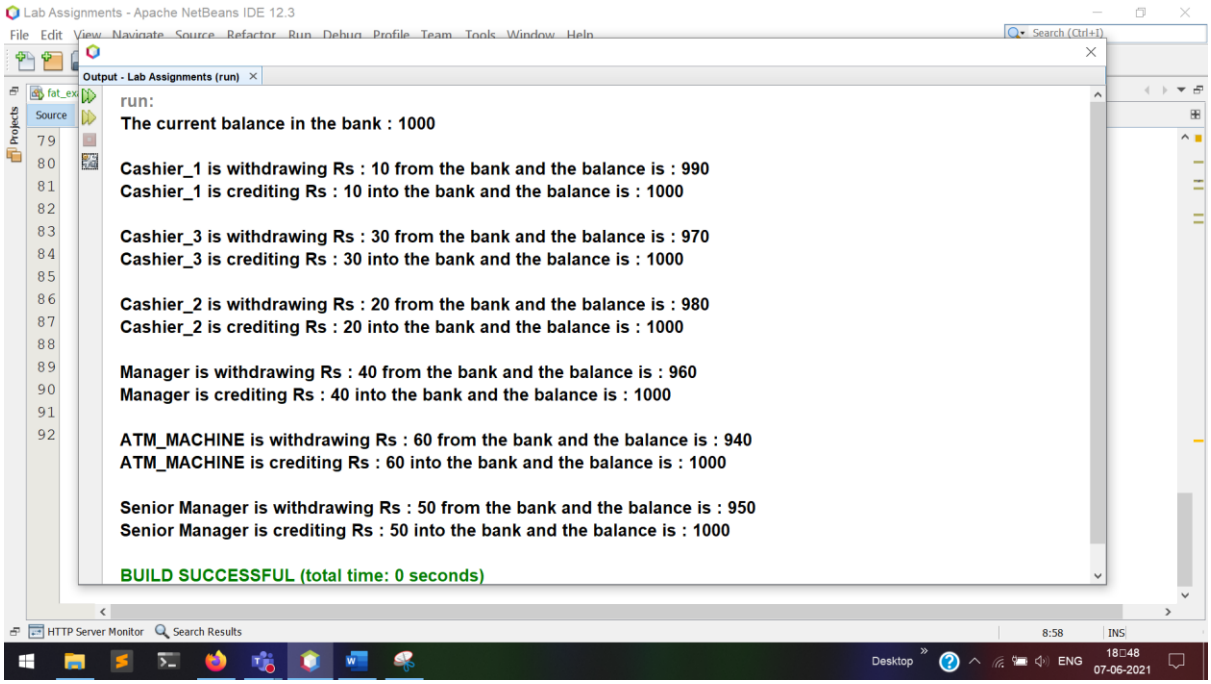
Code screen shot




Lab Assignments - Apache NetBeans IDE 12.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
fat_exam.java
Source History
class banking_operations extends Thread {
String thread_name;
int amount;
bank obj_bank; // main class
banking_operations(String name_from_main,int amount_main, bank obj_main) {
this.thread_name = name_from_main;
this.obj_bank = obj_main;
this.amount = amount_main;
}
public void operations() { // calling the credit() and debit()
this.obj_bank.debit(this.thread_name,this.amount);
this.obj_bank.credit(this.thread_name,this.amount);
}
@Override
// to do both the process credit and debit operations by the cashier and the executives
public void run() {
this.operations();
}
HTTP Server Monitor
Search Results
71:1
INS Windows (CR...
Desktop
ENG
18:45
07-06-2021


Lab Assignments - Apache NetBeans IDE 12.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
fat_exam.java
Source History
public class fat_exam {
public static void main(String[] args) throws Exception{
System.out.println("The current balance in the bank : 1000\n");
bank obj_bank = new bank();
banking_operations cashier_1 = new banking_operations("Cashier_1",10,obj_bank);
banking_operations cashier_2 = new banking_operations("Cashier_2",20,obj_bank);
banking_operations cashier_3 = new banking_operations("Cashier_3",30,obj_bank);
banking_operations manager = new banking_operations("Manager",40,obj_bank);
banking_operations senior_manager = new banking_operations("Senior Manager",50,obj_bank);
banking_operations atm_machine = new banking_operations("ATM_MACHINE",60,obj_bank);
// starting the thread
cashier_1.start();
cashier_2.start();
cashier_3.start();
try{ Thread.sleep(200);
manager.start();
senior_manager.start();
atm_machine.start();
}catch(Exception e){};
}
}
HTTP Server Monitor
Search Results
45:1
INS Windows (CR...
Desktop
ENG
18:46
07-06-2021

Output


Lab Assignments - Apache NetBeans IDE 12.3
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Output - Lab Assignments (run)
run:
The current balance in the bank : 1000
Cashier_1 is withdrawing Rs : 10 from the bank and the balance is : 990
Cashier_1 is crediting Rs : 10 into the bank and the balance is : 1000
Cashier_3 is withdrawing Rs : 30 from the bank and the balance is : 970
Cashier_3 is crediting Rs : 30 into the bank and the balance is : 1000
Cashier_2 is withdrawing Rs : 20 from the bank and the balance is : 980
Cashier_2 is crediting Rs : 20 into the bank and the balance is : 1000
Manager is withdrawing Rs : 40 from the bank and the balance is : 960
Manager is crediting Rs : 40 into the bank and the balance is : 1000
ATM_MACHINE is withdrawing Rs : 60 from the bank and the balance is : 940
ATM_MACHINE is crediting Rs : 60 into the bank and the balance is : 1000
Senior Manager is withdrawing Rs : 50 from the bank and the balance is : 950
Senior Manager is crediting Rs : 50 into the bank and the balance is : 1000
BUILD SUCCESSFUL (total time: 0 seconds)
HTTP Server Monitor
Search Results
8:58
INS
Desktop
ENG
18:48
07-06-2021

Actual Code

```
package lab.assignments;


class bank {
        int Netbalance=1000;   // Currently the bank has the balance amout=10,000


        // debitting the amount from the bank ( balance money reduces)
        synchronized public void debit(String person_name,int amount)   {
            this.Netbalance = this.Netbalance-amount;
            System.out.println(person_name + " is withdrawing Rs : "+ amount + " from the bank"
                + " and the balance is : " + Netbalance);


          }


        // creditting the amount into the bank (balance money increases)
        synchronized public void credit(String person_name,int amount)  {
            this.Netbalance = this.Netbalance + amount;
            System.out.println(person_name + " is crediting Rs : "+ amount + " into the bank"
                + " and the balance is : " + Netbalance + "\n");
        }
}


class banking_operations extends Thread {
        String thread_name;
        int amount;
        bank obj_bank;  // main class


        banking_operations(String name_from_main,int amount_main, bank obj_main) {
            this.thread_name = name_from_main;
            this.obj_bank = obj_main;
            this.amount = amount_main;
```

```
    }

        public void operations() {  // calling the credit() and debit()
            this.obj_bank.debit(this.thread_name,this.amount);
            this.obj_bank.credit(this.thread_name,this.amount);
        }

        @Override
        // to do both the process credit and debit operations by the cashier and the executives
        public void run() {
            this.operations();
        }
}


public class fat_exam {
   public static void main(String[] args) throws Exception{
        System.out.println("The current balance in the bank : 1000\n");
        bank obj_bank = new bank();
       banking_operations cashier_1 = new banking_operations("Cashier_1",10,obj_bank);
       banking_operations cashier_2 = new banking_operations("Cashier_2",20,obj_bank);
       banking_operations cashier_3 = new banking_operations("Cashier_3",30,obj_bank);

       banking_operations manager = new banking_operations("Manager",40,obj_bank);
       banking_operations senior_manager = new banking_operations("Senior
Manager",50,obj_bank);
       banking_operations atm_machine = new banking_operations("ATM_MACHINE",60,obj_bank);

       // starting the thread
       cashier_1.start();
       cashier_2.start();
```

```
        cashier_3.start();


         try{ Thread.sleep(200);
              manager.start();
              senior_manager.start();
              atm_machine.start();
        }catch(Exception e){};
    }
}
```