Activity - 2

Create a class account that maintains acc_no, name, and balance. Perform deposit, withdrawal and statement print operations.

Use subclasses loan account, credit card, and its relevant variables, functions related to it.

The class must be able to throw a INSUFFICIENT FUND exception when a user is trying to withdraw a amount less the minimum balance.

Over-view of Assignment-2

```java
  1        package lab.assignments;
  2
  ⊚        class Account {
  4            String acc_no;
  5            String name;
  6            int balance;
  7            int salary;
  8
  9    ⊞  │ Account(String user_acc_no, String user_name, int user_balance) {...5 lines }
 14
 15    ⊞  │ void Statement() {...6 lines }
 21
 22                      // Defining a user defined exception
 23    ⊞  │ void Deposit_negative(int amount) throws Exception  {...8 lines }
 31
 32    ⊞  │ void Deposit(int amount) {...8 lines }
 40
 41                      // Defining a user defined exception
 42    ⊞  │ void Withdraw_insufficient(int amount) throws Exception {...8 lines }
 50
 51    ⊞  │ void Withdraw(int amount) {...8 lines }
 59
 60    }
 61
 62    class Loan extends Account {
 63        int loan_user_salary , loan_user_amount;
 64
 65    ⊞      Loan(String user_acc_no, String user_name, int user_balance,int user_salary,int loan_amount) {...5 lines }
 70
 71    ⊞      void Loan_Exception() throws Exception {...8 lines }
 79
 80    ⊞      void Loan() {...8 lines }
 88    }
 89
 90    class Credit_card extends Account {
 91        int credit_user_salary , credit_user_amount;
 92
 93    ⊞      Credit_card(String user_acc_no, String user_name, int user_balance,int user_salary,int loan_amount) {...5 lines }
 98
 99    ⊞      void Credit_Swipe_Exception() throws Exception {...8 lines }
107
108    ⊞          void Credit_Swipe() {...9 lines }
117    }
118
119    public class Assignment_2 {
120    ⊞      public static void main(String args[]) {...24 lines }
144    }
145
```

Account class

```java
class Account {
        String acc_no;
        String name;
        int balance;
        int salary;

    Account(String user_acc_no, String user_name, int user_balance) {  // general details for opening a bank account
        this.acc_no  = user_acc_no;
        this.name    = user_name;
        this.balance = user_balance;
    }

    void Statement() {...6 lines }

                    // Defining a user defined exception
    void Deposit_negative(int amount) throws Exception  {
      if (amount<0)  {  // amount should not be -ve
        throw new Exception ("Amount cannot be negative");
      }
        else {
            this.balance = this.balance + amount;
        }
    }

    void Deposit(int amount) {
        try {
            Deposit_negative(amount);
        }
        catch(Exception obj) {
            System.out.println("Exception from Deposit_negative : " + obj);
        }
    }

                    // Defining a user defined exception
    void Withdraw_insufficient(int amount) throws Exception {
      if (   (amount < this.balance) && ((this.balance-amount)<5000)) {
        throw new Exception ("In-sufficient balance");
      }
        else {
            this.balance = this.balance - amount;
        }
    }

    void Withdraw(int amount) {
        try {
            Withdraw_insufficient(amount);
        }
        catch (Exception obj){
            System.out.println("Exception from Withdraw_insufficient" + obj);
        }
    }

}
```

Loan class

```java
class Loan extends Account {
    int loan_user_salary , loan_user_amount;

    Loan(String user_acc_no, String user_name, int user_balance,int user_salary,int loan_amount) {
        super(user_acc_no, user_name, user_balance);
        this.loan_user_salary = user_salary;
        this.loan_user_amount = loan_amount;
    }

    void Loan_Exception() throws Exception {
        if (   (this.loan_user_amount) > (this.loan_user_salary + ((0.75)*this.loan_user_salary))   ) {
            throw new Exception("You Loan request is not accepted");
        }
    }

    void Loan_amount() {
        try {
            if (   (this.loan_user_amount) <= (this.loan_user_salary + ((0.75)*this.loan_user_salary))   ) {
                System.out.println("SUCCESS --> You Loan request is accepted");
            }
            else {
                Loan_Exception();
            }
        }
        catch(Exception obj) {
            System.out.println("EXCEPTION : " + obj);
        }
    }
}
```

Credit card class

```java
 92    class Credit_card extends Account {
 93        int credit_user_salary , credit_user_amount;
 94
 95        Credit_card(String user_acc_no, String user_name, int user_balance,int user_salary,int credit_amount) {
 96        super(user_acc_no, user_name, user_balance);
 97        this.credit_user_salary = user_salary;
 98        this.credit_user_amount = credit_amount;
 99    }
100
101        void Credit_Swipe_Exception() throws Exception {
102            if (  (this.credit_user_amount) > (this.credit_user_salary + ((0.25)*this.credit_user_salary))  ) {
103                    throw new Exception("You cannot swipe your quoted amount");
104            }
105        }
106
107        void Credit_Swipe() {
108            try {
109                if (  (this.credit_user_amount) <= (this.credit_user_salary + ((0.25)*this.credit_user_salary))  ) {
110                    System.out.println("SUCCESS --> You can swipe your quoted amount");
111                }
112                else        {
113                        Credit_Swipe_Exception();
114                }
115            }
116
117        catch(Exception obj) {
118            System.out.println("EXCEPTION : " + obj);
119        }
120    }
121 }
122
```

Main method

```
119    public class Assignment_2 {
120        public static void main(String args[]) {
121
122            // getting details from the user to open a bank account
123            Account obj1 = new Account("19MID1000","GilChrist",5000);
124            obj1.Deposit(1000);
125            obj1.Withdraw(3000);
126
127            // generating a statement
128            obj1.Statement();
129
130            // getting details from the user to apply for a loan
131            Loan obj2 = new Loan("19MID0020","Prashanth",50_000,1_00_000,1_75_000);
132            obj2.Loan();
133
134            // generating a statement
135            obj2.Statement();
136
137            // getting details from the user to swipe the credit card
138            Credit_card obj3 = new Credit_card("19MID0017","Mothish",50_000,1_00_000,1_25_000);
139            obj3.Credit_Swipe();
140
141            // generating a statement
142            obj3.Statement();
143        }
144    }
```

Depositing and Withdrawing Procedures

Depositing negative amount

```
122            // getting details from the user to open a bank account
123            Account obj1 = new Account("19MID1000","GilChrist",10000);
               obj1.Deposit(-1000);
125            obj1.Withdraw(3000);
126
127            // generating a statement
128            obj1.Statement();
129            /*
```

```
Output ×

WinterSemester-2021 - C:\WinterSemester-2021 ×    Debugger Console ×    Lab_Assignments (run-single) ×

init:
Deleting: C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\built-jar.properties
deps-jar:
Updating property file: C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\built-jar.proper
Compiling 1 source file to C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\classes
compile-single:
run-single:
Exception from Deposit_negative : java.lang.Exception: Amount cannot be negative
Your Bank details
Account no : 19MID1000
User Name : GilChrist
Balance : 7000
BUILD SUCCESSFUL (total time: 1 second)
```

Withdrawing amount which is greater than minimum balance

```
122        // getting details from the user to open a bank account
123        Account obj1 = new Account("19MID1000","GilChrist",10000);
124        obj1.Deposit(1000);
           obj1.Withdraw(7000);
126
127        // generating a statement
128        obj1.Statement();
129        /*
```
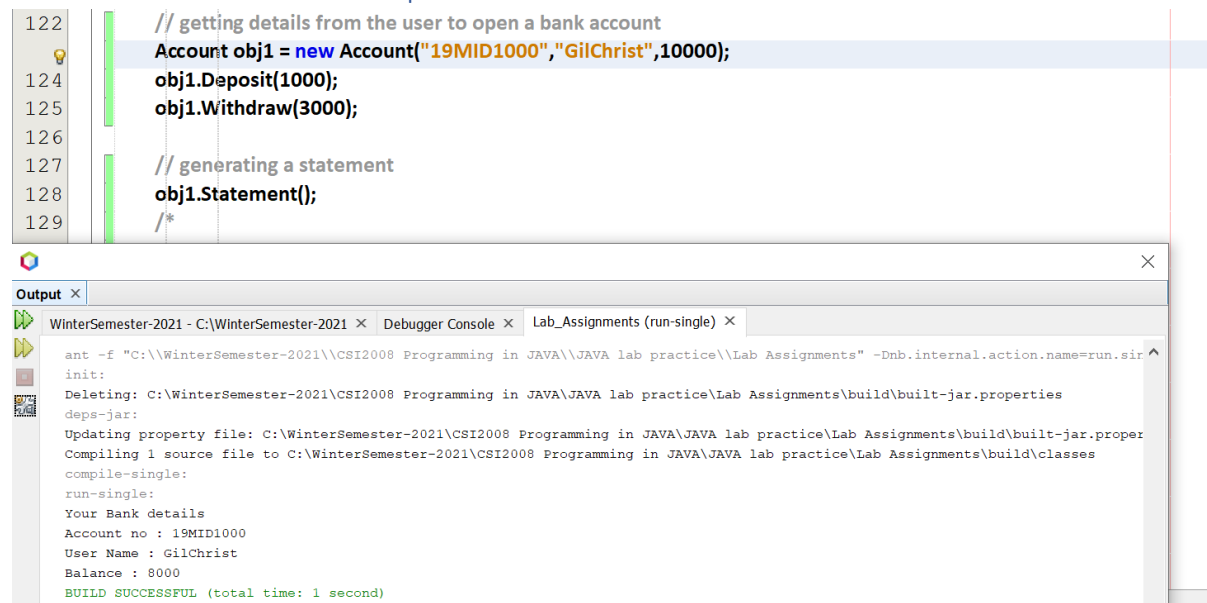
Output ✕

WinterSemester-2021 - C:\WinterSemester-2021 ✕   Debugger Console ✕   Lab_Assignments (run-single) ✕

```
init:
Deleting: C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\built-jar.properties
deps-jar:
Updating property file: C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\built-jar.proper
Compiling 1 source file to C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\classes
compile-single:
run-single:
Exception from Withdraw_insufficientjava.lang.Exception: In-sufficient balance
Your Bank details
Account no : 19MID1000
User Name : GilChrist
Balance : 11000
BUILD SUCCESSFUL (total time: 1 second)
```

With the above two exception

```
123        Account obj1 = new Account("19MID1000","GilChrist",10000);
           obj1.Deposit(-1000);
125        obj1.Withdraw(7000);
126
127        // generating a statement
128        obj1.Statement();
129        /*
```

Output ✕

WinterSemester-2021 - C:\WinterSemester-2021 ✕   Debugger Console ✕   Lab_Assignments (run-single) ✕

```
Deleting: C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\built-jar.properties
deps-jar:
Updating property file: C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\built-jar.proper
Compiling 1 source file to C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\classes
compile-single:
run-single:
Exception from Deposit_negative : java.lang.Exception: Amount cannot be negative
Exception from Withdraw_insufficientjava.lang.Exception: In-sufficient balance
Your Bank details
Account no : 19MID1000
User Name : GilChrist
Balance : 10000
BUILD SUCCESSFUL (total time: 0 seconds)
```
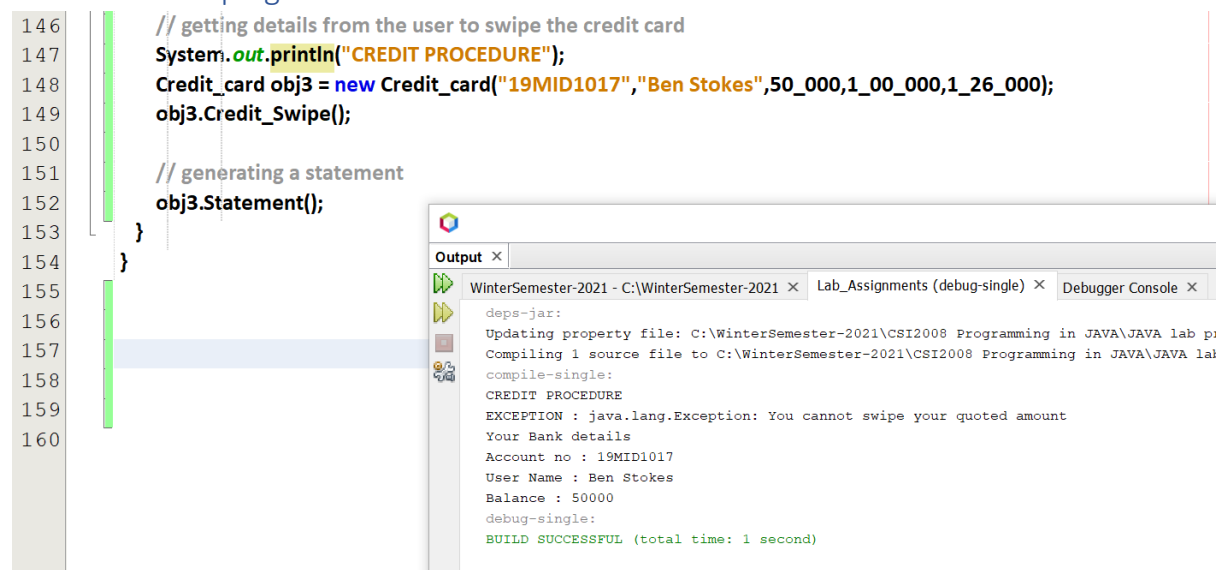
Without the above two exception

```
122        // getting details from the user to open a bank account
           Account obj1 = new Account("19MID1000","GilChrist",10000);
124        obj1.Deposit(1000);
125        obj1.Withdraw(3000);
126
127        // generating a statement
128        obj1.Statement();
129        /*
```

Credit card and Loan Amount Procedures

Credit card → <= (salary + $1/4^{th}$ * salary) → Approve

Loan → <= (salary + $3/4^{th}$ * salary) → Approve

Credit card Swiping more amount

```
146        // getting details from the user to swipe the credit card
147        System.out.println("CREDIT PROCEDURE");
148        Credit_card obj3 = new Credit_card("19MID1017","Ben Stokes",50_000,1_00_000,1_26_000);
149        obj3.Credit_Swipe();
150
151        // generating a statement
152        obj3.Statement();
153    }
154 }
155
156
157
158
159
160
```
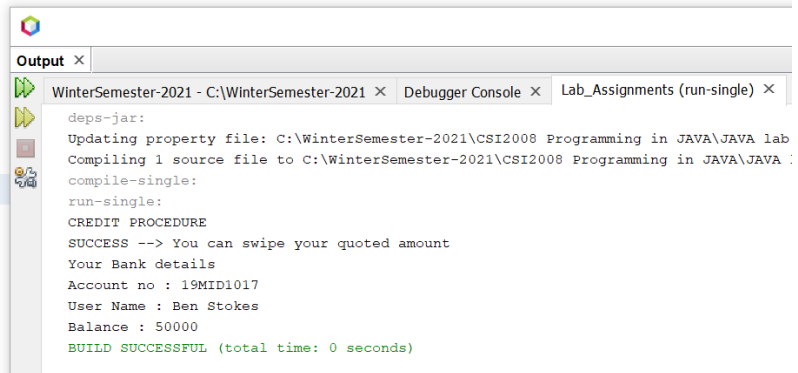
Credit card Swiping less amount

```
146            // getting details from the user to swipe the credit card
147            System.out.println("CREDIT PROCEDURE");
148            Credit_card obj3 = new Credit_card("19MID1017","Ben Stokes",50_000,1_00_000,1_24_000);
149            obj3.Credit_Swipe();
150
151            // generating a statement
152            obj3.Statement();
153        }
154    }
155
156
157
158
159
160
```
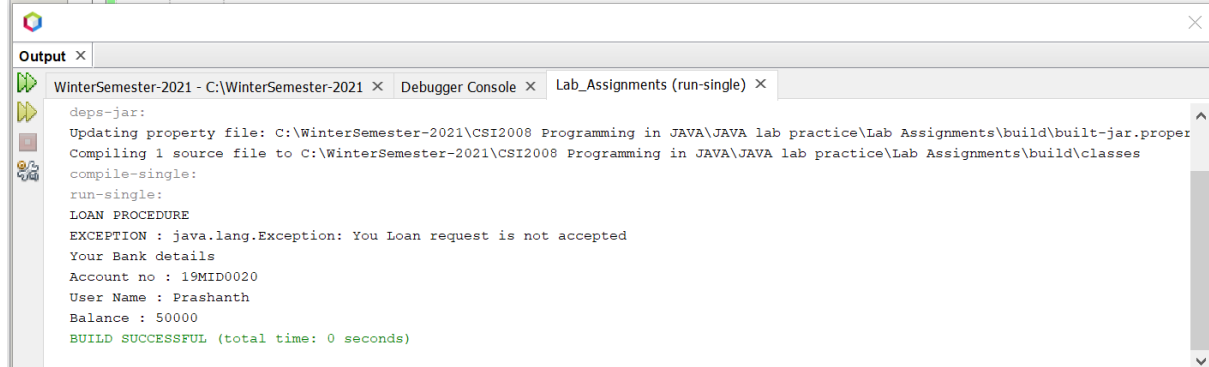
```
Output ×

WinterSemester-2021 - C:\WinterSemester-2021 ×   Debugger Console ×   Lab_Assignments (run-single) ×

deps-jar:
Updating property file: C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab
Compiling 1 source file to C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA
compile-single:
run-single:
CREDIT PROCEDURE
SUCCESS --> You can swipe your quoted amount
Your Bank details
Account no : 19MID1017
User Name : Ben Stokes
Balance : 50000
BUILD SUCCESSFUL (total time: 0 seconds)
```

Applying for more loan amount

```
136                    // getting details from the user to apply for a loan
137            System.out.println("LOAN PROCEDURE");
               Loan obj2 = new Loan("19MID0020","Prashanth",50_000,1_00_000,1_76_000);
139            obj2.Loan_amount();
140
141            // generating a statement
142            obj2.Statement();
```

```
Output ×

WinterSemester-2021 - C:\WinterSemester-2021 ×   Debugger Console ×   Lab_Assignments (run-single) ×

deps-jar:
Updating property file: C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\built-jar.proper
Compiling 1 source file to C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\classes
compile-single:
run-single:
LOAN PROCEDURE
EXCEPTION : java.lang.Exception: You Loan request is not accepted
Your Bank details
Account no : 19MID0020
User Name : Prashanth
Balance : 50000
BUILD SUCCESSFUL (total time: 0 seconds)
```

Applying for less loan amount

```
136              // getting details from the user to apply for a loan
137        System.out.println("LOAN PROCEDURE");
138        Loan obj2 = new Loan("19MID0020","Prashanth",50_000,1_00_000,1_74_000);
139        obj2.Loan_amount();
140
141        // generating a statement
142        obj2.Statement();
```
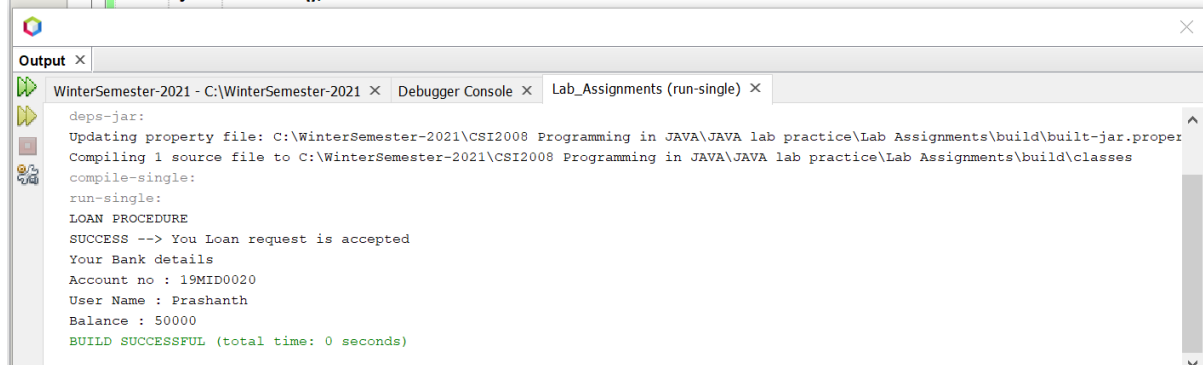
```
Output ×

WinterSemester-2021 - C:\WinterSemester-2021 ×   Debugger Console ×   Lab_Assignments (run-single) ×

deps-jar:
Updating property file: C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\built-jar.proper
Compiling 1 source file to C:\WinterSemester-2021\CSI2008 Programming in JAVA\JAVA lab practice\Lab Assignments\build\classes
compile-single:
run-single:
LOAN PROCEDURE
SUCCESS --> You Loan request is accepted
Your Bank details
Account no : 19MID0020
User Name : Prashanth
Balance : 50000
BUILD SUCCESSFUL (total time: 0 seconds)
```

Without the above two exception

```
136              // getting details from the user to apply for a loan
137        System.out.println("LOAN PROCEDURE");
138        Loan obj2 = new Loan("19MID0020","Prashanth",50_000,1_00_000,1_74_000);
139        obj2.Loan_amount();
140
141        // generating a statement
142        obj2.Statement();
143
144        // getting details from the user to swipe the credit card
145        System.out.println("CREDIT PROCEDURE");
146        Credit_card obj3 = new Credit_card("19MID1017","Ben Stokes",50_000,1_00_000,1_24_000);
147        obj3.Credit_Swipe();
148
149        // generating a statement
150        obj3.Statement();
151        }
152    }
153
154
155
156
```
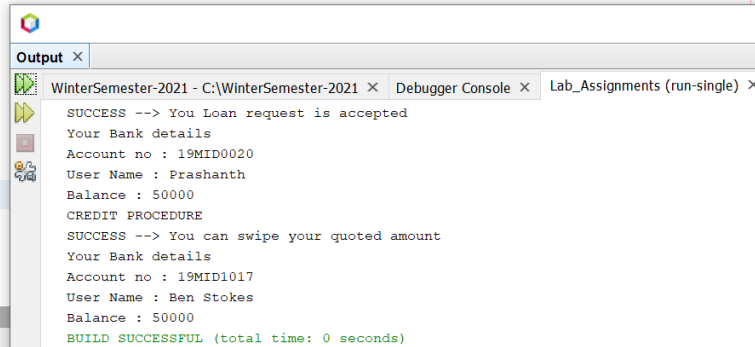
```
Output ×

WinterSemester-2021 - C:\WinterSemester-2021 ×   Debugger Console ×   Lab_Assignments (run-single) ×

SUCCESS --> You Loan request is accepted
Your Bank details
Account no : 19MID0020
User Name : Prashanth
Balance : 50000
CREDIT PROCEDURE
SUCCESS --> You can swipe your quoted amount
Your Bank details
Account no : 19MID1017
User Name : Ben Stokes
Balance : 50000
BUILD SUCCESSFUL (total time: 0 seconds)
```

With the above two exception

```
136              // getting details from the user to apply for a loan
137         System.out.println("LOAN PROCEDURE");
138         Loan obj2 = new Loan("19MID0020","Prashanth",50_000,1_00_000,1_76_000);
139         obj2.Loan_amount();
140
141         // generating a statement
142         obj2.Statement();
143
144         // getting details from the user to swipe the credit card
145         System.out.println("CREDIT PROCEDURE");
146         Credit_card obj3 = new Credit_card("19MID1017","Ben Stokes",50_000,1_00_000,1_26_000);
147         obj3.Credit_Swipe();
148
149         // generating a statement
150         obj3.Statement();
151     }
152 }
153
154
155
156
```

lab.assignments.Assignment_2 > main >

Output ×

WinterSemester-2021 - C:\WinterSemester-2021 ×   Debugger Console ×   Lab_Assignments (run-single) ×

```
EXCEPTION : java.lang.Exception: You Loan request is not accepted
Your Bank details
Account no : 19MID0020
User Name : Prashanth
Balance : 50000
CREDIT PROCEDURE
EXCEPTION : java.lang.Exception: You cannot swipe your quoted amount
Your Bank details
Account no : 19MID1017
User Name : Ben Stokes
Balance : 50000
BUILD SUCCESSFUL (total time: 0 seconds)
```

Activity - 3

Write a synchronization program for ticket booking process in theatre. Consider three counters for booking and have num_tickets has a common variable holding available tickets. When counter1 is using it others should not access to variable num_tickets.

Code:

```java
1      package lab_assignments;
2
3      class Ticket_counter {
4
5        synchronized public void Printing() {
6             Counter.count++;
7             System.out.println(Thread.currentThread().getName() + "  ticket_no : " + (Counter.count-1));
8        }
9
10       public void booking() {   // ctustomer is printing (i.e actual booking) the ticket and then sleeping
         while (Counter.count<=50) {  Printing() ;  try{ Thread.sleep(1000);}  catch (Exception e) { };    }
12       }
13
14     }
15
16     class Counter extends Thread {
17       Ticket_counter obj1 = new Ticket_counter();
18       public static int count=1;
19
20       Counter(String name)   {  super(name);     }
         public void run()        {  obj1.booking();    }
22     }
23
24     public class Assignment_3_Synchronisation {
25       public static void main(String args[]) {
           Ticket_counter obj1 = new Ticket_counter();
27
28         Counter th1 = new Counter("Counter-1");
29         Counter th2 = new Counter("Counter-2");
30         Counter th3 = new Counter("Counter-3");
31
32         // At a time all the coustomers entering the ticket counter
33         th1.start();        th2.start();       th3.start();
34
           try { th1.join();    th2.join();    th3.join(); } catch(InterruptedException e) {} ;
36       }
37     }
```

Output:

Total number of tickets → 5

```
Updating property file: F:\2) Second Year 2020-2021\WinterSemester-2021\CSI2008 Programming in JAVA\Lab_assignments\build\built-jar.properties
Compiling 1 source file to F:\2) Second Year 2020-2021\WinterSemester-2021\CSI2008 Programming in JAVA\Lab_assignments\build\classes
compile-single:
run-single:
Counter-2    ticket_no : 2
Counter-1    ticket_no : 1
Counter-3    ticket_no : 3
Counter-2    ticket_no : 4
Counter-1    ticket_no : 4
Counter-3    ticket_no : 5
Counter-2    ticket_no : 6
Counter-3    ticket_no : 7
Counter-1    ticket_no : 8
Counter-2    ticket_no : 9
Counter-3    ticket_no : 10
Counter-1    ticket_no : 11
Counter-3    ticket_no : 12
Counter-2    ticket_no : 13
Counter-1    ticket_no : 14
Counter-3    ticket_no : 15
Counter-2    ticket_no : 16
Counter-1    ticket_no : 17
Counter-3    ticket_no : 18
Counter-2    ticket_no : 19
Counter-1    ticket_no : 20
Counter-3    ticket_no : 21
Counter-2    ticket_no : 22
Counter-1    ticket_no : 23
Counter-3    ticket_no : 24
Counter-2    ticket_no : 25
Counter-1    ticket_no : 26
Counter-3    ticket_no : 27
Counter-2    ticket_no : 28
Counter-1    ticket_no : 29
Counter-3    ticket_no : 30
Counter-2    ticket_no : 31
Counter-1    ticket_no : 32
Counter-3    ticket_no : 33
Counter-2    ticket_no : 34
Counter-1    ticket_no : 35
Counter-3    ticket_no : 36
Counter-2    ticket_no : 37
Counter-1    ticket_no : 38
Counter-3    ticket_no : 39
Counter-2    ticket_no : 40
Counter-1    ticket_no : 41
Counter-3    ticket_no : 42
Counter-2    ticket_no : 43
Counter-1    ticket_no : 44
Counter-3    ticket_no : 45
Counter-2    ticket_no : 46
Counter-1    ticket_no : 47
Counter-3    ticket_no : 48
Counter-2    ticket_no : 49
Counter-1    ticket_no : 50
BUILD SUCCESSFUL (total time: 17 seconds)
```