***Generate an One Time Password***

For secure transaction a one-time password (OTP) is a password used in all transactions that is valid for only one login session or transaction, on a computer system or other digital device.

***Generation Guidelines:***

input1 (String) => Name of the account holder

input2 (Number) => Account Number

input3 (String) => Type of the Account

output1 => Generated OTP

***Assumptions:***

1. Name of the account holder’s length should be minimum of 2.

2. Number of digits of Account Number is fixed as 11.

3. Account type should be any one of given below { Savings / Current / Loan }

***Rules:***

1. The length of the OTP is exactly 8

2. The OTP starts with first two characters of the account type (Case sensitive)

3. The third character is first digit of the account number

4. The fourth character is the sum of first five digits of account number (if the sum exceeds single digit then sum of digits should be done to form a single digit. Repeat if necessary)

5. The fifth character is the first character of the account holder’s name

6. The sixth character is the sum of sixth to tenth digit positions of the account number (if the sum exceeds single digit then sum of digits should be done to form a single digit. Repeat if necessary)

7. The seventh character is the last character of the account holder’s name

8. The eighth character is the last digit of the account number

9. All characters should be same case of the input string

***Example 1:***

input1 => Kavi

input2 => 12345678925

input3 => Savings

output1 => Sa16K5i5

|  |  |  |
| --- | --- | --- |
| **Character Steps** | **Rules Applied** | **Output** |
| First & Second Character | **Sa**vings | Sa |
| Third Character | **1**2345678925 | Sa1 |
| Fourth Character | **12345**678925  1 + 2 + 3 + 4 + 5 = 15  1 + 5 = 6 | Sal6 |
| Fifth Character | Kavi | Sal6K |
| Sixth Character | 12345**67892**5  6 + 7 + 8 + 9 + 2 = 32  3 + 2 = 5 | Sal6K5 |
| Seventh Character | **Kavi** | Sal6K5i |
| Eight Character | 12345678925 | Sal6K5i5 |
|  |  |  |

Program

**import** java.util.Scanner;

**public** **class** Generate\_password {

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

// **TODO** Auto-generated method stub

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter the account holder name:");

String acc\_name=sc.next();

System.***out***.println("Enter the account number:");

String acc\_no=sc.next();

System.***out***.println("Enter the account type(Savings/Current/Loan):");

String acc\_type=sc.next();

String output="";

**int** arr[]=**new** **int**[11];

**if**(acc\_type.equalsIgnoreCase("Savings")||acc\_type.equalsIgnoreCase("Current")||acc\_type.equalsIgnoreCase("Loan"))

{

output+=acc\_type.substring(0,2)+acc\_no.substring(0,1);

**int** sum=0;

**if**(acc\_no.length()==11)

{

**for**(**int** i=0;i<acc\_no.length();i++)

{

arr[i]=Character.*getNumericValue*(acc\_no.charAt(i));

}

**for**(**int** i=0;i<5;i++)

sum+=arr[i];

**int** ans = 0;

**if**(sum>9)

{

ans=*sumdigit*(sum);

output+=ans+acc\_name.substring(0,1);

}

**else**

output+=sum+acc\_name.substring(0,1);

**int** sum1=0;

**int** ans1=0;

**for**(**int** i=5;i<10;i++)

sum1+=arr[i];

**if**(sum1>9)

{

ans1=*sumdigit*(sum1);

output+=ans1+acc\_name.substring(acc\_name.length()-1)+arr[10];

}

**else**

output+=sum1+acc\_name.substring(acc\_name.length()-1)+arr[10];

System.***out***.println(output);

}

**else**

System.***out***.println("Enter the valid account number");

}

**else**

System.***out***.println("Enter a valid account type");

}

**private** **static** **int** sumdigit(**int** sum) {

// **TODO** Auto-generated method stub

**int** n=sum;

sum=0;

**while**(n!=0) {

sum+=n%10;

n/=10;

}

//System.out.println(sum);

**if**(sum>9)

{

System.***out***.println("asghdsgfysd");

sum=*sumdigit*(sum);

}

**return** sum;

}

}

Output

**Test Case 1:**

input1 => GoodnamE

input2 => 61741978349

input3 => Current

output1 => Cu61G4E9

**Test Case 2:**

input1 => Balakrishnan

input2 => 73150000312

input3 => Loan

output1 =>Lo77B4n2

**Test Case 3:**

input1 => Selvam

input2 => 31651234576

input3 => Current

output1 =>Cu37S3m6

**Test Case 4:**

input1 => veronicA

input2 => 22233555667

input3 => saving

output1 => sa23v9A7

**Test Case 5:**

input1 => wasimakram

input2 => 73172223456

input3 => lOan

output1 => lO72w7m6

**Test Case 6:**

input1 => CAanad

input2 => 12123454567

input3 => Current

output1 => Cu19C6d7