# Counting Objects :

```
 package com.css.corexercise;
//Counting No.Of Objects in class
public class CountObjects {
static int count=0;
CountObjects()
{
        count++;
}
public static void main(String args[])
{
        CountObjects o1=new CountObjects();
        CountObjects o2=new CountObjects();
        System.out.println("No.Of Objects :" + count);
        }
}
```

OUTPUT :

No.Of Objects : 2

# App4Bank :

```
package com.css.app4bank.dao;


import com.css.app4bank.dto.BankAccount;


public interface IBankServiceProvider {


	public boolean createAccount(String accountName);

	public BankAccount checkAccount(String accountNo);

	public double getBalance(String accountNo);

	public boolean depositMoney(String accountNo,double amount);

	public boolean withdrawMoney(String accountNo,double amount);

	public boolean transferMoney(String fromAccountNo,String toAccountNo,double
amount);

}
```

BANKACCOUNT : ;

```
package com.css.app4bank.dto;
```

```
public class BankAccount {


	private String accountNo;

	private String accountName;

	private double balance;


	private static int lastAssignedNo;

	static {

		lastAssignedNo=0;

	}


	public BankAccount(String accountName) {

		super();

		this.balance=1000;

		this.accountNo=Integer.toString(lastAssignedNo+1);
//""+lastAssignedNo+1

		lastAssignedNo++;

		this.accountName = accountName;

	}

	public BankAccount(String accountNo, String accountName) {

		super();

		this.accountNo = accountNo;

		this.accountName = accountName;
```

```
            this.balance=1000.00;
      }
      public String getAccountNo() {
            return accountNo;
      }
      public void setAccountNo(String accountNo) {
            this.accountNo = accountNo;
      }
      public String getAccountName() {
            return accountName;
      }
      public void setAccountName(String accountName) {
            this.accountName = accountName;
      }
      public double getBalance() {
            return balance;
      }
      public void setBalance(double balance) {
            this.balance = balance;
      }
      public BankAccount(String accountNo, String accountName, double
balance) {
            super();
            this.accountNo = accountNo;
```

```
            this.accountName = accountName;

            this.balance = balance;

      }

      public BankAccount() {

            super();

            // TODO Auto-generated constructor stub

      }

      @Override

      public String toString() {

            return "BankAccount [accountNo=" + accountNo + ",
accountName=" + accountName + ", balance=" + balance + "]";

      }
```

BANKSERVICEPROVIDER : :

```
package com.css.app4bank.dto;


import java.util.Set;


import com.css.app4bank.dao.IBankServiceProvider;


public class Bank implements IBankServiceProvider {

      private String IFSC;

      private String bankName;
```

```
//private BankAccount[] accounts;
private Set<BankAccount> accounts;



public Bank(String iFSC, String bankName) {
	super();
	IFSC = iFSC;
	this.bankName = bankName;
}

public String getIFSC() {
	return IFSC;
}

public void setIFSC(String iFSC) {
	IFSC = iFSC;
}

public String getBankName() {
	return bankName;
}

public void setBankName(String bankName) {
```

```
		this.bankName = bankName;
	}




	public Set<BankAccount> getAccounts() {
		return accounts;
	}


	public void setAccounts(Set<BankAccount> accounts) {
		this.accounts = accounts;
	}


	public Bank() {
		super();
		// TODO Auto-generated constructor stub
	}


	public Bank(String iFSC, String bankName, Set<BankAccount>
accounts) {
		super();
		IFSC = iFSC;
		this.bankName = bankName;
		this.accounts = accounts;
```

```
	}




	@Override
	public String toString() {
		return "Bank [IFSC=" + IFSC + ", bankName=" + bankName + ",
accounts=" + accounts + "]";
	}



	@Override
	public BankAccount checkAccount(String accountNo) {
		BankAccount foundAccount = null;
		for (BankAccount account : this.accounts) {
			if (account.getAccountNo().equals(accountNo)) {
				foundAccount = account;
				break;
			}
		}
		return foundAccount;
	}



	@Override
	public double getBalance(String accountNo) {
```

```
		double balance = 0.0;
		BankAccount foundAccount = this.checkAccount(accountNo);
		if (foundAccount != null)
			balance = foundAccount.getBalance();
		return balance;
	}

	@Override
	public boolean depositMoney(String accountNo, double amount) {
		boolean depositFlag = false;
		BankAccount foundAccount = this.checkAccount(accountNo);
		if (foundAccount != null) {
			double newBalance = foundAccount.getBalance() +
amount;
			foundAccount.setBalance(newBalance);
			depositFlag = true;
		}
		return depositFlag;
	}

	@Override
	public boolean withdrawMoney(String accountNo, double amount) {
		boolean withdrawFlag = false;
		BankAccount foundAccount = this.checkAccount(accountNo);
```

```
            if (foundAccount != null) {
                  if (foundAccount.getBalance() >= amount) {
                        double newBalance = foundAccount.getBalance() -
amount;
                        foundAccount.setBalance(newBalance);
                        withdrawFlag = true;
                  }
            }
            return withdrawFlag;
      }

      @Override
      public boolean transferMoney(String fromAccountNo, String
toAccountNo, double amount) {
            boolean transferFlag=false;

            boolean withdrawFlag=this.withdrawMoney(fromAccountNo,
amount);
            if(withdrawFlag) {
                  boolean depositFlag=this.depositMoney(toAccountNo,
amount);
                  if(depositFlag)
                        transferFlag=true;
                  else
                        this.depositMoney(fromAccountNo,amount);
```

```
            }

            return transferFlag;
      }

      @Override
      public boolean createAccount(String accountName) {
            boolean createAccountFlag=false;
            BankAccount newAccount=new BankAccount(accountName);
            this.accounts.add(newAccount);
            if(this.accounts.contains(newAccount))
                  createAccountFlag=true;

            return createAccountFlag;
      }

}
```

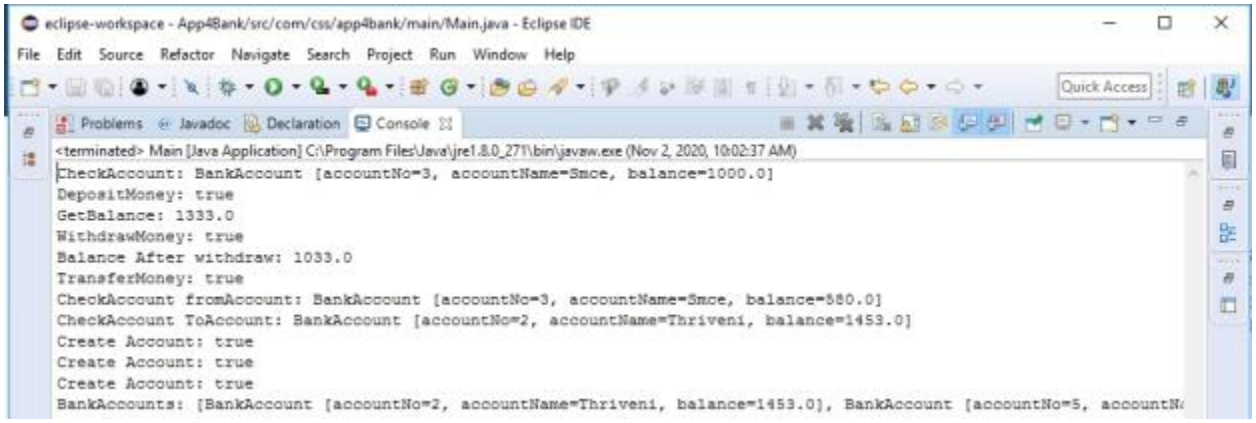
MAIN ::

```
package com.css.app4bank.main;
```

```
import java.util.HashSet;

import java.util.Set;


import com.css.app4bank.dto.Bank;

import com.css.app4bank.dto.BankAccount;


public class Main {


      public static void main(String[] args) {


            Set<BankAccount> accounts=new HashSet<BankAccount>();


            BankAccount ba1=new BankAccount("Vardhani");

            BankAccount ba2=new BankAccount("Thriveni");

            BankAccount ba3=new BankAccount("Smce");


            accounts.add(ba1);

            accounts.add(ba2);

            accounts.add(ba3);


            Bank axis=new Bank("IND001","Indian Bank");

            axis.setAccounts(accounts);

            BankAccount
foundAccount=axis.checkAccount(Integer.toString(3));
```

```
            System.out.println("CheckAccount:
"+foundAccount.toString());

            System.out.println("DepositMoney:
"+axis.depositMoney(Integer.toString(3),333));

            System.out.println("GetBalance:
"+axis.getBalance(Integer.toString(3)));

            System.out.println("WithdrawMoney:
"+axis.withdrawMoney(Integer.toString(3),300));

            System.out.println("Balance After withdraw:
"+axis.getBalance(Integer.toString(3)));

            System.out.println("TransferMoney:
"+axis.transferMoney(Integer.toString(3), Integer.toString(2), 453));

            System.out.println("CheckAccount fromAccount:
"+axis.checkAccount(Integer.toString(3)).toString());

            System.out.println("CheckAccount ToAccount:
"+axis.checkAccount(Integer.toString(2)).toString());

            System.out.println("Create Account:
"+axis.createAccount("Anant"));

            System.out.println("Create Account:
"+axis.createAccount("Varan"));

            System.out.println("Create Account:
"+axis.createAccount("Bharath"));

            System.out.println("BankAccounts:
"+axis.getAccounts().toString());

      }



}
```

OUTPUT :




## 9th Excersice ::

```
import java.util.*;

public class Employee {

String empname;

public static void main (String args[])

{


HashSet<String> set = new HashSet<String>();
set.add("thriveni");
```

```
set.add("vardhani");

set.add("Syamala");

set.add("vani");


for(into count:empname)


{


count=0;

 if(set.Contains(empname)) count++;

if(Employee.getempname().equals(this.empname))

return empid;

}

}}
```

Product :

```
public class Store


{


public static void main (String args[])
```

```
{

ArrayList<String > product =new ArrayList<String>();
product.add("soap");
product.add("cloths");
product.add("electronics"); for(String
i:product) System.out.println(i);
}

}
```

## JAVA PROGRAM TO STORE TODAY'S DATE AND DATE OF BIRTH

```
public class Date{

public static void main (String args[]){
LocalDate today =LocalDate now();
LocalDate birthday = LocalDate .of(1999,month.July,24); Period
p= Period.between (birthday,today);
System.out.println("Period.getdays());
System.out.println("Period.getmonths());
System.out.println("Period.getyears());
}
}
```

# Java Assignment