```
}

//Deethya Makonahalli's student account that takes methods from account and stores
new variables which aim to calculate the cummulative balance of a stuent

  public class StudentAccount extends Account{;

    //initialising the field account number
    private String accountnumber;

    //initialising a field of type LibraryAccount for LibraryAccount
    private LibraryAccount library;

    //initialising a field of type CreditAccount for TuitionAccount
    private CreditAccount tuitionAccount;

    //initialising a field of type CreditAccount for DiningAccount
    private CreditAccount diningAccount;

    //initialising the field name
    private String name;

    //initialising the field address
    private String address;

    //initialising the field payment for credit method
    private double payment;

    //initialising the field balance
    private double balance;

  // Constructor for Student account to input accountNumber and account holder name
   public StudentAccount(String accountNumber, String name){
     super(accountNumber);
     accountNumber = accountNumber;
     this.name = name;
}

  // Setting the Name of the account holder(the student)
   public void setName(String name){
     this.name = name;
  }

  // Method to access the student Name

   public String getName(){
     return name;
}

  // Method to set the Address of the student

   public void setAddress(String address){
     this.address = address;
}

  // Method to access the Address of the student

   public String getAddress(){
     return address;
}
```

```
  // Set the LibraryAccount to a value of type LibraryAccount

   public void setLibraryAccount(LibraryAccount library){
     this.library = library;

}

  // Method to access the Library Account value

   public LibraryAccount getLibraryAccount(){
     return library;
}

  // Set a value for TuitionAccount of type CreditAccount

   public void setTuitionAccount(CreditAccount tuitionAccount){
     this.tuitionAccount = tuitionAccount;

}

  // Method to access the Tuition account

   public CreditAccount getTuitionAccount(){
     return tuitionAccount;
}

  // Method to set a value for DiningAccount of type CreditAccount

   public void setDinningAccount(CreditAccount diningAccount){
     this.diningAccount = diningAccount;

}

  // Method to access the Dining Account

   public CreditAccount getDiningAccount(){
     return diningAccount;
}

  // Method to check if the account exists for each type of account and adjust the
Balance after decreasing each account in the order Tuition account, Dining Account,
and Library Account

  @Override
   public double getBalance(){
     if (tuitionAccount != null) {
    }if (library != null){
    }if(diningAccount != null){
  }
      this.setBalance((this.getLibraryAccount()).getBalance() +
(this.getTuitionAccount()).getBalance() + (this.getDiningAccount()).getBalance()
       - super.getBalance() );
      return balance;
}

  // Method to Change the balance by the input charge

  @Override
   public void charge(double c){
     if(c - this.getBalance() >=0){
        this.setBalance(getTuitionAccount().getBalance() + (c - balance));
     }else {
      setBalance(c - this.getBalance());
   }
 }

  //Method to adjust each account by the input payment in the order of tuition,
dining then library

  @Override
   public void credit(double payment){
     this.payment = payment;

      if (tuitionAccount != null) {

      if (getTuitionAccount().getBalance() - payment ==
tuitionAccount.getMonthlyPayment()){
        tuitionAccount.setBalance(getTuitionAccount().getBalance()-payment);
   }
     }else if ((getDiningAccount() != null) && getTuitionAccount().getBalance()>0
&& (getDiningAccount().getBalance()-payment >= diningAccount.getMonthlyPayment())){
      diningAccount.setBalance(getDiningAccount().getBalance()-payment);

    } else if ((library != null) && getDiningAccount().getBalance() > 0 &&
getLibraryAccount().getBalance()-payment >= 0){
        library.setBalance(getLibraryAccount().getBalance()-payment);

    } else if (getLibraryAccount().getBalance() + getTuitionAccount().getBalance()
+ getDiningAccount().getBalance()>0){
            super.setBalance(getLibraryAccount().getBalance() +
getTuitionAccount().getBalance() + getDiningAccount().getBalance());
           }
    }
}
```