JAVA PROGRAMMING

JAVA INTERFACE

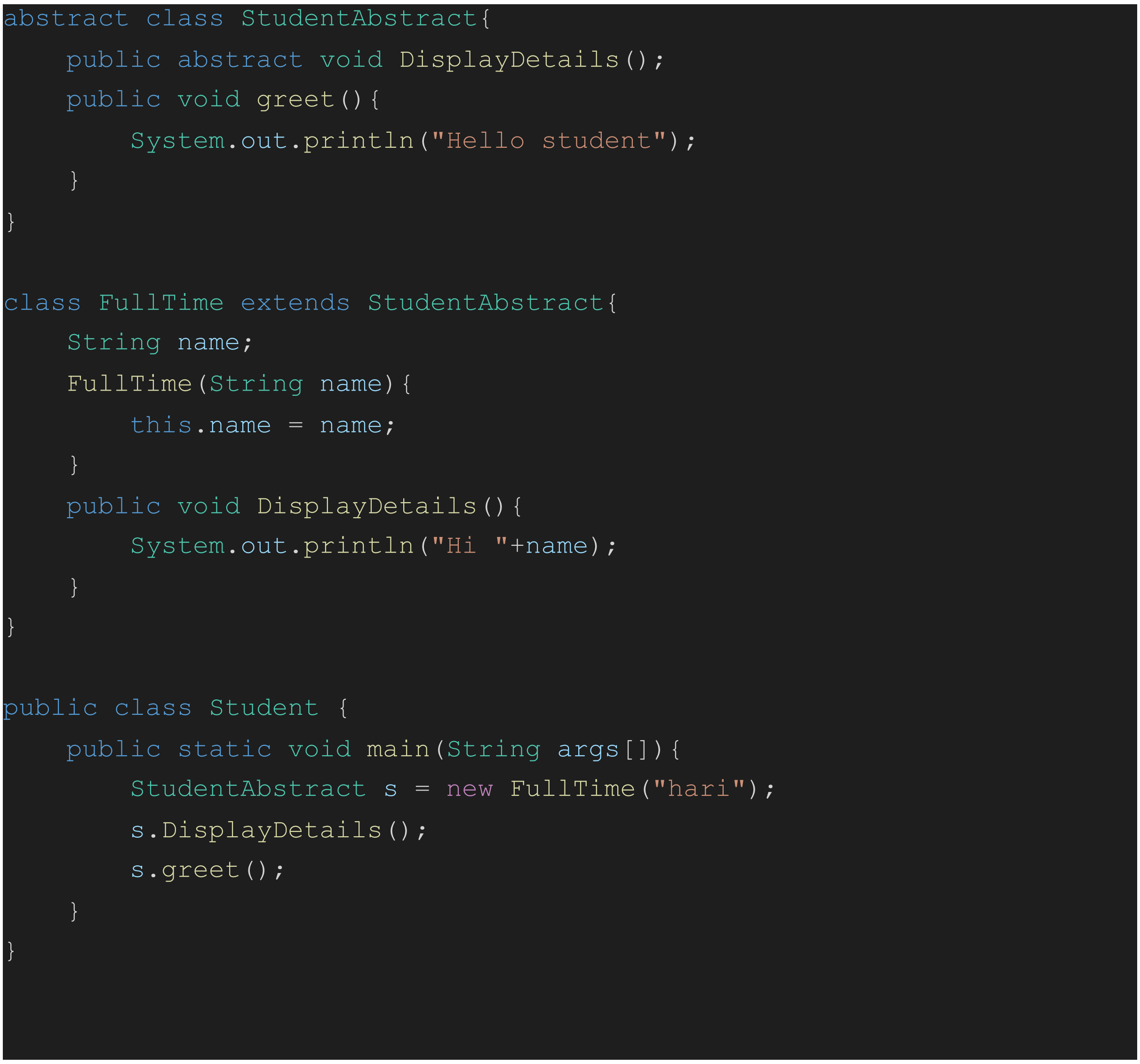
2018503030

HARIPRASANTH M S

MO BATCH

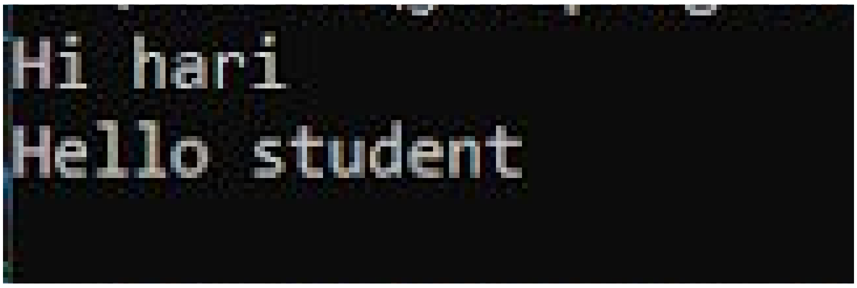
1) 1.create an abstract class name student. Inside the student class you maintain concrete method and abstract method.
Create a class called full-time and define the abstract methods
try to create an object method for students and refer to the full time object and access the abstract method

SOURCE CODE:



```
abstract class StudentAbstract{
    public abstract void DisplayDetails();
    public void greet(){
        System.out.println("Hello student");
    }
}

class FullTime extends StudentAbstract{
    String name;
    FullTime(String name){
        this.name = name;
    }
    public void DisplayDetails(){
        System.out.println("Hi "+name);
    }
}

public class Student {
    public static void main(String args[]){
        StudentAbstract s = new FullTime("hari");
        s.DisplayDetails();
        s.greet();
    }
}
```

OUTPUT:



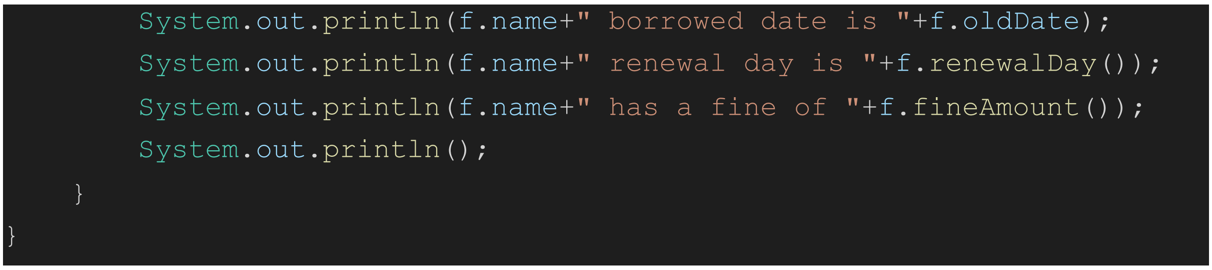
```
Hi hari
Hello student
```

2) Create an interface for library application. add atleast two interface methods for renewal of books and renewal day calculation and amount calculation create two classes as student and faculty. try to implement the interface methods in the classes.

SOURCE CODE:

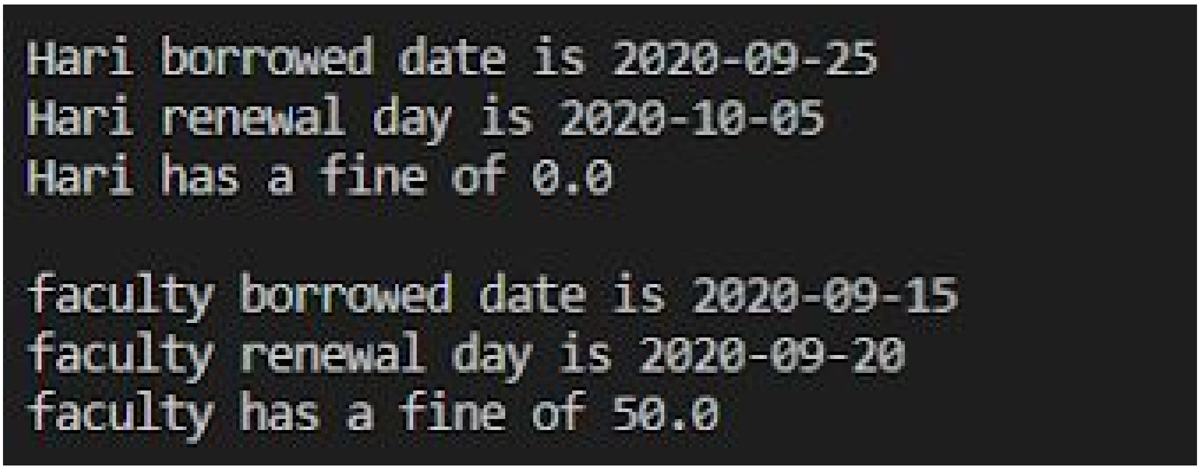
```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;

interface LibraryInterface{
    public String renewalDay();
    public double fineAmount();
}

class Student implements LibraryInterface{
    String name;
    String oldDate;
    SimpleDateFormat sdf;
    Calendar c;

    Student(String name){
        this.name = name;
        sdf = new SimpleDateFormat("yyyy-MM-dd");
        c = Calendar.getInstance();
        oldDate = sdf.format(c.getTime());
    }
    public String renewalDay(){
        Calendar cal = Calendar.getInstance();
        try{
            cal.setTime(sdf.parse(oldDate));
         }catch(ParseException e){
            e.printStackTrace();
        }
        cal.add(Calendar.DAY_OF_MONTH, 10);
        return sdf.format(cal.getTime());
    }
    public double fineAmount(){
        Calendar cal = Calendar.getInstance();
        long time = ((cal.getTimeInMillis() -
c.getTimeInMillis())/(1000*86400)-14);
        return (time>0?time*10:0);
    }
}
```

```
class Faculty implements LibraryInterface{
    String name;
    String oldDate;
    SimpleDateFormat sdf;
    Calendar c;

    Faculty(String name){
        this.name = name;
        sdf = new SimpleDateFormat("yyyy-MM-dd");
        c = Calendar.getInstance();
        c.add(Calendar.DAY_OF_MONTH, -10);
        oldDate = sdf.format(c.getTime());
    }
    public String renewalDay(){
        Calendar cal = Calendar.getInstance();
        try{
            cal.setTime(sdf.parse(oldDate));
         }catch(ParseException e){
            e.printStackTrace();
        }
        cal.add(Calendar.DAY_OF_MONTH, -10);
        return sdf.format(cal.getTime());
    }
    public double fineAmount(){
        Calendar cal = Calendar.getInstance();
        long time = ((cal.getTimeInMillis() -
c.getTimeInMillis())/(1000*86400));
        return (time>0?time*10:0);
    }
}



public class Library {
    public static void main(String args[]){
        Student s = new Student("Hari");
        Faculty f = new Faculty("faculty");
        System.out.println();
        System.out.println(s.name+" borrowed date is "+s.oldDate);
        System.out.println(s.name+" renewal day is "+s.renewalDay());
        System.out.println(s.name+" has a fine of "+s.fineAmount());
        System.out.println();
```



```
        System.out.println(f.name+" borrowed date is "+f.oldDate);
        System.out.println(f.name+" renewal day is "+f.renewalDay());
        System.out.println(f.name+" has a fine of "+f.fineAmount());
        System.out.println();
    }
}
```

OUTPUT:



```
Hari borrowed date is 2020-09-25
Hari renewal day is 2020-10-05
Hari has a fine of 0.0

faculty borrowed date is 2020-09-15
faculty renewal day is 2020-09-20
faculty has a fine of 50.0
```