ASSIGNMENT 4

1. Create a class Student with StudentId, Name, MobileNo, Address, Course. Write getters and

setters for all the data members, Validate Name - should not be greater than 15 characters.

MobileNo should have compulsorily 10 digits. Write a method CalculateFee which returns the

fee depending on course taken.

Write child classes FastTrackBatchStudent, CorporateBatchStudent, WeekendBatch Student,

CorporateWeekendBatchStudent which overrides CalculateFee method and returns appropriate

salary.

Student.java

```
public abstract class Student {



      private int studentId;

      private String studentName;

      private long  studentMobile;

      private String studentAddress;

      private String studentCourse;


    public Student(int studentId, String studentName, long
studentMobile, String studentAddress, String studentCourse) {

           super();

           this.studentId = studentId;
    this.studentName = studentName;

    this.studentMobile = studentMobile;

    this.studentAddress = studentAddress;

    this.studentCourse = studentCourse;

}


public int getStudentId() {

    return studentId;

}


public void setStudentId(int studentId) {

    this.studentId = studentId;

}


public String getStudentName() {

    return studentName;

}


public void setStudentName(String studentName) {

    this.studentName = studentName;

}


public long getStudentMobile() {

    return studentMobile;

}
```

```java
    public void setStudentMobile(long studentMobile) {
        this.studentMobile = studentMobile;
    }

    public String getStudentAddress() {
        return studentAddress;
    }

    public void setStudentAddress(String studentAddress) {
        this.studentAddress = studentAddress;
    }

    public String getStudentCourse() {
        return studentCourse;
    }

    public void setStudentCourse(String studentCourse) {
        this.studentCourse = studentCourse;
    }

    abstract int CalculateFee(String course);
}
```

FAST TRACK BATCH

```java
public class FastTrackBatch extends Student {
    public FastTrackBatch(int studentId, String studentName, long
studentMobile, String studentAddress,
            String studentCourse) {
        super(studentId, studentName, studentMobile, studentAddress,
studentCourse);
        // TODO Auto-generated constructor stub
    }


    @Override
    int CalculateFee(String course) {
        // TODO Auto-generated method stub
        if(course.equals("java")){
            return 25000;
        }
        else if(course.equals("dotnet")){
            return 15000;
        }
        else{
            return 10000;
        }
    }


}
```

WEEKENDBATCH

```
public class WeekendBatch extends FastTrackBatch {


    public WeekendBatch(int studentId, String studentName, long
studentMobile, String studentAddress,

            String studentCourse) {

        super(studentId, studentName, studentMobile, studentAddress,
studentCourse);

        // TODO Auto-generated constructor stub

    }

    @Override

    int CalculateFee(String course) {

        // TODO Auto-generated method stub

        if(course.equals("java")){

            return 50000;

        }

        else if(course.equals("dotnet")){

            return 25000;

        }

        else{

            return 15000;

        }

    }


}
```

Corporate Batch

```
public class CorporateBatch extends FastTrackBatch {


    public CorporateBatch(int studentId, String studentName, long
studentMobile, String studentAddress,
            String studentCourse) {
        super(studentId, studentName, studentMobile, studentAddress,
studentCourse);
        // TODO Auto-generated constructor stub
    }
    @Override
    int CalculateFee(String course) {
        // TODO Auto-generated method stub
        if(course.equals("java")){
            return 35000;
        }
        else if(course.equals("dotnet")){
            return 25000;
        }
        else{
            return 20000;
        }
    }


}
```

CorporateWeekendBatch

```
public class CorporateWeekendBatch extends WeekendBatch {

    public CorporateWeekendBatch(int studentId, String studentName, long studentMobile, String studentAddress,
            String studentCourse) {
        super(studentId, studentName, studentMobile, studentAddress, studentCourse);
        // TODO Auto-generated constructor stub
    }
    @Override
    int CalculateFee(String course) {
        // TODO Auto-generated method stub
        if(course.equals("java")){
            return 80000;
        }
        else if(course.equals("dotnet")){
            return 40000;
        }
        else{
            return 25000;
        }
    }
}
```

StudentClient

```
public class StudentClient {


    public static void main(String[] args) {

        // TODO Auto-generated method stub

        Student st=new
FasttrackBatchStudent(100,"Divya",9895389380l,"usa","java");

        String studCourse=st.getStudentCourse();

        int sfees=st.CalculateFee(studCourse);


    System.out.println(st.getStudentId()+"\t"+st.getStudentName()+"\t
"+st.getStudentMobile()+"\t"+st.getStudentAddress()+"\t"+st.getStudent
Course()+"\t"+"student fees is "+sfees);

        Student st1=new
WeekendBatchStudent(101,"Jayakrishnan",9895389381l,"usa","dotnet");

        String stud1Course=st1.getStudentCourse();

        int swfees=st1.CalculateFee(stud1Course);


    System.out.println(st1.getStudentId()+"\t"+st1.getStudentName()+"
\t"+st1.getStudentMobile()+"\t"+st1.getStudentAddress()+"\t"+st1.getSt
udentCourse()+"\t"+"student fees is "+swfees);

        Student st2=new
CorporateBatchStudent(102,"Arun",9895389382l,"london","python");

        String stud2Course=st2.getStudentCourse();

        int scfees=st2.CalculateFee(stud2Course);
```

```
     System.out.println(st2.getStudentId()+"\t"+st2.getStudentName()+"
\t"+st2.getStudentMobile()+"\t"+st2.getStudentAddress()+"\t"+st2.getSt
udentCourse()+"\t"+"student fees is "+scfees);

          Student st3=new
CorporateWeekendBatchStudent(103,"veena",9895389383l,"london","dotnet"
);

          String stud3Course=st1.getStudentCourse();

          int scwfees=st3.CalculateFee(stud3Course);


     System.out.println(st3.getStudentId()+"\t"+st3.getStudentName()+"
\t"+st3.getStudentMobile()+"\t"+st3.getStudentAddress()+"\t"+st3.getSt
udentCourse()+"\t"+"student fees is "+scwfees);


}

}
```

2. Create a class Employee which has the following

```
int EmpId;

double Sal = 0;

double Basic;

double Allowance;

double Deducions;

string FirstName;

string LastName;

string Address;

string Pincode;
```

Make Salary as ReadOnly

Write getters and setters realname which returns firstname+lastname realname should not be less than 0 characters

Write a method CalcSalary to calculate salary

Sal= Basic + Allowance-Deductions

Write a parameterised constructor and save the above values

Write a class PartTimeEmployee extending Employee class and override CalcSalary with

different implementation

Write a class FullTimeEmployee

Write a class NightShiftEmployees -- extra allowance for cab

Write a class Manager -- who is a full time employee and works on night shifts

Solution:

Employee.java

```
public class Employee {
      int EmpId;
      private double Sal = 0;
      double Basic;
      double Allowance;
      double Deductions;
      String FirstName;
      String LastName;
      String Address;
      String Pincode;
      String Realname;
      public int getEmpId() {
          return EmpId;
      }
      public void setEmpId(int empId) {
          EmpId = empId;
      }
      public double getBasic() {
          return Basic;
      }
      public void setBasic(double basic) {
          Basic = basic;
      }
      public double getAllowance() {
          return Allowance;
      }
      public void setAllowance(double allowance) {
          Allowance = allowance;
      }
      public double getDeductions() {
          return Deductions;
      }
      public void setDeductions(double deductions) {
          Deductions = deductions;
      }
      public String getFirstName() {
          return FirstName;
```

```java
}
public void setFirstName(String firstName) {
      FirstName = firstName;
}
public String getLastName() {
      return LastName;
}
public void setLastName(String lastName) {
      LastName = lastName;
}
public String getAddress() {
      return Address;
}
public void setAddress(String address) {
      Address = address;
}
public String getPincode() {
      return Pincode;
}
public void setPincode(String pincode) {
      Pincode = pincode;
}
public String getRealname() {
      return (FirstName+" "+LastName);
}
```

```
    public void setRealname(String realname) {
        Realname = realname;
    }
    double CalcSalary(double basicSalary,double allw,double dedu){
        double totalSal=basicSalary+allw-dedu;
        return totalSal;
    }
    public void setSal(double sal) {
        Sal = sal;
    }
    public double getSal() {
        return Sal;
    }
    public Employee(int empId, double sal, double basic, double
allowance, double deductions, String firstName,
            String lastName, String address, String pincode,
String realname) {
        super();
        EmpId = empId;
        Sal = sal;
        Basic = basic;
        Allowance = allowance;
        Deductions = deductions;
        FirstName = firstName;
        LastName = lastName;
```

```
		Address = address;

		Pincode = pincode;

		Realname = realname;

	}

}
```

PartTimeEmployee.java

```
public class PartTimeEmployee extends Employee {


	public PartTimeEmployee(int empId, double sal, double basic,
double allowance, double deductions, String firstName,

			String lastName, String address, String pincode,
String realname) {

		super(empId, sal, basic, allowance, deductions, firstName,
lastName, address, pincode, realname);

		// TODO Auto-generated constructor stub

	}


	double CalcSalary(double basicSalary,double allw,double dedu,int
hours){


		double totalSal=basicSalary*hours+allw-dedu;

		return totalSal;


	}

}
```

FullTimeEmployee.java

```
public class FullTimeEmployee extends PartTimeEmployee {
	public FullTimeEmployee(int empId, double sal, double basic,
double allowance, double deductions, String firstName,
			String lastName, String address, String pincode,
String realname) {
		super(empId, sal, basic, allowance, deductions, firstName,
lastName, address, pincode, realname);
		// TODO Auto-generated constructor stub
	}

	double CalcSalary(double basicSalary,double allw,double dedu){

		int hours=45;
		double totalSal=basicSalary*hours+allw-dedu;
		return totalSal;

	}
}
```

NightShiftEmployee.java

```
public class NightShiftEmployees extends FullTimeEmployee {

	public NightShiftEmployees(int empId, double sal, double basic,
double allowance, double deductions
```

```
                ,String firstName, String lastName, String address, 
String pincode, String realname) {
          super(empId, sal, basic, allowance, deductions, firstName, 
lastName, address, pincode, realname);
          // TODO Auto-generated constructor stub
     }
     double CalcSalary(double basicSalary, double allw, double 
dedu,double cabAllw) {
          // TODO Auto-generated method stub
          return basicSalary+allw-dedu+cabAllw;
     }
}
```

Manager.java

```
public class Manager extends FullTimeEmployee {


     public Manager(int empId, double sal, double basic, double 
allowance, double deductions, String firstName,
                String lastName, String address, String pincode, 
String realname) {
          super(empId, sal, basic, allowance, deductions, firstName, 
lastName, address, pincode, realname);
          // TODO Auto-generated constructor stub
     }
     double CalcSalary(double basicSalary, double allw, double 
dedu,double cabAllw) {
          // TODO Auto-generated method stub
```

```
        int hours=45;

        return basicSalary*hours+allw-dedu+cabAllw;

    }

}
```

EmployeeClient.java

```
import java.util.Scanner;


public class EmployeeClient {


    public static void main(String[] args) {

        // TODO Auto-generated method stub

        PartTimeEmployee em=new PartTimeEmployee(100,0,100,300,150,"Divya","Prakash","usa","95035",null);

        System.out.println("The real name of the employee is : "+em.getRealname());

        Scanner sc=new Scanner(System.in);

        System.out.println("enter the number of hours worked by part time employee");

        int hrs=sc.nextInt();

        double salary=em.CalcSalary(em.getBasic(),em.getAllowance(),em.getDeductions(),hrs);

        em.setSal(salary);

        System.out.println("The salary of "+em.getRealname()+"is Rs."+em.getSal());
```

```
           Manager em1=new
Manager(103,0,100,300,150,"Jayakrishnan","Muraleedharan","usa","95035"
,null);

           System.out.println("The real name of the employee is :
"+em.getRealname());

           Scanner sc1=new Scanner(System.in);

           System.out.println("enter cab allowance for night shift
employees");

           double allw1=sc1.nextInt();

           double
salary1=em1.CalcSalary(em1.getBasic(),em1.getAllowance(),em1.getDeduct
ions(),allw1);

           em1.setSal(salary1);

           System.out.println("The salary of "+em1.getRealname()+"is
Rs."+em1.getSal());

}

}
```

3. Create two interfaces IMobile and ITelephone and have common functionalities like Dial in

them.

Create a class called as Mobile inherited from both the interfaces

Create child classes for Mobile - Samsung, Nokia, IPhone which has the following members and

Functions

Data Members: IEMICode, IsSingleSIM, Processor, SIMCard, MobileNo

Member Functions: ConnectBlueTooth, Dial, GetIEMICode, GetWIFIConnection, Receive, SendMessage

Generate another level of inheritance like Samsung S5, Nokia Lumis625 and so on...

Solution:

Mobile.java

```
interface ITelephone{
    void dial();
}
interface  IMobile extends ITelephone{
    void ConnectBlueTooth();
    void GetIEMICode(int code);
    void GetWIFIConnection();
    void Receive();
    void SendMessage();

}
public class Mobile implements IMobile,ITelephone{
    int IEMICode;
    long IsSingleSIM;
    String Processor;
    long SIMCard;
    long MobileNo;

    public int getIEMICode() {
        return IEMICode;
    }

    public void setIEMICode(int iEMICode) {
        IEMICode = iEMICode;
    }

    public long getIsSingleSIM() {
        return IsSingleSIM;
    }

    public void setIsSingleSIM(long isSingleSIM) {
        IsSingleSIM = isSingleSIM;
    }

    public String getProcessor() {
        return Processor;
    }

    public void setProcessor(String processor) {
        Processor = processor;
    }

    public long getSIMCard() {
        return SIMCard;
    }
```

```java
    public void setSIMCard(long sIMCard) {
        SIMCard = sIMCard;
    }

    public long getMobileNo() {
        return MobileNo;
    }

    public void setMobileNo(long mobileNo) {
        MobileNo = mobileNo;
    }

    public Mobile(int iEMICode, long isSingleSIM, String processor, long sIMCard, long mobileNo) {
        super();
        IEMICode = iEMICode;
        IsSingleSIM = isSingleSIM;
        Processor = processor;
        SIMCard = sIMCard;
        MobileNo = mobileNo;
    }

    @Override
```

```
}

public void dial() {
        // TODO Auto-generated method stub
        System.out.println("Dial the number");

}

@Override
public void ConnectBlueTooth() {
        // TODO Auto-generated method stub
        System.out.println("Connecting to bluetooth");
}

@Override
public void GetIEMICode(int code) {
        // TODO Auto-generated method stub
        System.out.println("The IEMICode is "+code);

}

@Override
public void GetWIFIConnection() {
        // TODO Auto-generated method stub
        System.out.println("Connecting to Wifi ");
}
```

```
    @Override
    public void Receive() {
        // TODO Auto-generated method stub
        System.out.println("Message Received");

    }

    @Override
    public void SendMessage() {
        // TODO Auto-generated method stub
        System.out.println("The message has been sent");

    }
}
```

Samsung.java

```
public class Samsung extends Mobile {

    public Samsung(int iEMICode, long isSingleSIM, String processor, long sIMCard, long mobileNo) {
        super(iEMICode, isSingleSIM, processor, sIMCard, mobileNo);
        // TODO Auto-generated constructor stub
    }
}
```

Nokia.java

```
public class Nokia extends Mobile {


    public Nokia(int iEMICode, long isSingleSIM, String processor, long sIMCard, long mobileNo) {

        super(iEMICode, isSingleSIM, processor, sIMCard, mobileNo);

        // TODO Auto-generated constructor stub

    }

}
```

IPhone.java

```
public class IPhone extends Mobile {


    public IPhone(int iEMICode, long isSingleSIM, String processor, long sIMCard, long mobileNo) {

        super(iEMICode, isSingleSIM, processor, sIMCard, mobileNo);

        // TODO Auto-generated constructor stub

    }

}
```

SamsungS5.java

```
public class SamsungS5 extends Samsung {
	public SamsungS5(int iEMICode, long isSingleSIM, String processor, long sIMCard, long mobileNo) {

		super(iEMICode, isSingleSIM, processor, sIMCard, mobileNo);

		// TODO Auto-generated constructor stub

	}

}
```

Nokia Lumis625

```
public class NokiaLumis625 extends Nokia {


	public NokiaLumis625(int iEMICode, long isSingleSIM, String processor, long sIMCard, long mobileNo) {

		super(iEMICode, isSingleSIM, processor, sIMCard, mobileNo);

		// TODO Auto-generated constructor stub

	}

}
```

4. Create a parent class Account which has fields such as

AccNo, Name, MobileNumber and methods Deposit, Withdraw, GetBalance

Create a child class SavingsAccount where we have interestrate and a extra method AddInterest

Create a child class CurrentAccount where we have interestrate and a extra method AddInterest

Create a class CheckingAccount where we have an extra member NoOfFreeTransactions, when a

transaction is made increment TransactionCount till the number does not exceed

NoOfFreeTransactions. If the Count exceeds free transaction then deduct fees from your

balance.

Solution:

Account.java

```
public class Account {
    private int AccNo;
    private String Name;
    private long MobileNumber;
 double openingBalance=10000;


    public Account(int accNo, String name, long mobileNumber, double openingBalance) {
        super();
        AccNo = accNo;
        Name = name;
        MobileNumber = mobileNumber;
        this.openingBalance = openingBalance;
    }
    public int getAccNo() {
        return AccNo;
    }
    public void setAccNo(int accNo) {
        AccNo = accNo;
    }
    public String getName() {
        return Name;
```

```
}
public void setName(String name) {
    Name = name;
}
public long getMobileNumber() {
    return MobileNumber;
}
public void setMobileNumber(long mobileNumber) {
    MobileNumber = mobileNumber;
}
public double getOpeningBalance() {
    return openingBalance;
}
public void setOpeningBalance(int openingBalance) {
    this.openingBalance = openingBalance;
}
void deposit(double dAmount){
    System.out.println("Before Deposit openingBalance :"+openingBalance);
    openingBalance=openingBalance+dAmount;
    System.out.println("After Deposit openingBalance :"+openingBalance);

}
void withdraw(double wAmount){
    System.out.println("Before withdraw openingBalance :"+openingBalance);
```

```
        if(openingBalance>wAmount)
        {
            openingBalance=openingBalance-wAmount;
            System.out.println("After withdraw openingBalance :"+openingBalance);


        }
            else
            System.out.println("Sufficent funds are not avilable");
    }
    void getBalance(){
    System.out.println(" Balance of the Account :"+openingBalance);
    }
}
```

SavingsAccount

```
public class SavingAccount extends Account {



    private double annualInterestRate;


    public SavingAccount(int accNo, String name, long mobileNumber, int openingBalance, double annualInterestRate) {
        super(accNo, name, mobileNumber,openingBalance);
        this.annualInterestRate = annualInterestRate;
    }

    public double getAnnualInterestRate() {
        return annualInterestRate;
    }

    public void setAnnualInterestRate(double annualInterestRate) {
        this.annualInterestRate = annualInterestRate;
    }

    public double addInterest(double annualInterest){
        return getOpeningBalance()*(annualInterest/1200);
        //System.out.println("  of the Account :"+openingBalance);



    }
}
```

CurrentAccount

```
public class CurrentAccount extends SavingAccount {

    public CurrentAccount(int accNo, String name, long mobileNumber, int openingBalance,
double annualInterestRate) {
        super(accNo, name, mobileNumber, openingBalance, annualInterestRate);
```

```
		// TODO Auto-generated constructor stub
	}
}
```

CheckingAccount

```
public class CheckingAccount extends Account {
	int transactionCount=0;
	public CheckingAccount(int accNo, String name, long mobileNumber, int openingBalance, int transactionCount) {
		super(accNo, name, mobileNumber, openingBalance);
		this.transactionCount = transactionCount;
	}

	public CheckingAccount(int accNo, String name, long mobileNumber, int openingBalance) {
		super(accNo, name, mobileNumber, openingBalance);
		// TODO Auto-generated constructor stub

	}

	@Override
	void deposit(double dAmount) {
		// TODO Auto-generated method stub
		transactionCount++;
		super.deposit(dAmount);
	}
```

```
    @Override
    void withdraw(double wAmount) {
        transactionCount++;
        // TODO Auto-generated method stub
        super.withdraw(wAmount);
    }
    public void checking(){
        int transactionCounts = 0;
        final int NoOfFreeTransactions=3;
        final double FEE=2.50;

        if(transactionCounts>NoOfFreeTransactions){
            double totalFees=FEE*(transactionCounts-NoOfFreeTransactions);
            super.withdraw(totalFees);
        }
        transactionCounts=0;
    }
}
```