SUGGESTED SCHEMA HOT

RUBRIC TYPE SCHEMA

(a) Java source code.

[50 Marks]

	none	bad	average	good
Main Class				
Running properly with correct result (including correct calculation, decimal places and the usage of if-else statement)	0	2	4	6
Main class able to display name and position properly	0	1	2	3
Main class able to display every staff income and staff loan properly	0	2	4	6
Main class able to display total income, total loan and net income properly	0	1	2	4
CompanyStaff class				
Correct data member for CompanyStaff	0	1	2	3
Correct constructors for CompanyStaff (including correct implementation for composition)	0	1	2	3
Correct methods for CompanyStaff	0	3	6	10
Company, StaffIncome, StaffLoan classes				
Correct data member for Company, StaffIncome, StaffLoan	0	1	2	3
Correct constructors for Company, StaffIncome, StaffLoan	0	2	4	6
Correct methods for Company, StaffIncome, StaffLoan	0	2	4	6

```
public class Company {

  private CompanyStaff[] staff;
  private int numberOfStaff;

  public Company() {
           staff = new CompanyStaff[60];
           numberOfStaff = 0;
  }

  public void addEmployee(String name,String position) {
           staff[numberOfStaff] = new CompanyStaff(name,position);
           numberOfStaff++;
  }


  public CompanyStaff getStaff(int staff_index) {
           return staff[staff_index];
  }
  public int getNumOfStaff() {
          return numberOfStaff;
  }

}


public  class CompanyStaff {

 // Data Attributes
  private String  name;
  private String  position;

  private StaffIncome[] staffIncome;
  private int numberOfIncome;

  private StaffLoan[] staffLoan;
  private int numberOfLoan;

  public CompanyStaff(String name, String position) {
    this.name=name;
    this.position=position;

    staffIncome = new StaffIncome[5];
    numberOfIncome = 0;

    staffLoan = new StaffLoan[5];
    numberOfLoan = 0;
  }

  public String getName() {
    return name;
  }

  public String getPosition() {
    return position;
  }

  public void addStaffIncome(StaffIncome income) {
    staffIncome[numberOfIncome] = income;
    numberOfIncome++;
  }

  public StaffIncome getStaffIncome(int income_index) {
    return staffIncome[income_index];
  }

  public int getNumOfIncome() {
    return numberOfIncome;
  }

   public double getTotalIncome(){
      double totalIncome=0;
      for (int i=0;i<numberOfIncome;i++)
      {
          totalIncome=totalIncome+getStaffIncome(i).getAmount();
      }

      return totalIncome;
  }
```

```
   public void addStaffLoan(StaffLoan loan) {
    staffLoan[numberOfLoan] = loan;
    numberOfLoan++;
  }

  public StaffLoan getStaffLoan(int loan_index) {
    return staffLoan[loan_index];
  }
  public int getNumOfLoan() {
    return numberOfLoan;
  }

   public double getTotalLoan(){
      double totalLoan=0;
      for (int i=0;i<numberOfLoan;i++)
      {
          totalLoan=totalLoan+getStaffLoan(i).getAmount();
      }

      return totalLoan;
  }

   public  double getNetIncome()
    {
       double netIncome=0.0;

       netIncome=getTotalIncome()-getTotalLoan();
       return netIncome;
   }

}


public class StaffIncome {
    private String incomeType;
    private double amount;

  public StaffIncome() {
    this( "Unknown",0.0);
  }
  public StaffIncome(String type,double amount) {
      incomeType=type;
      this.amount = amount;
  }

  public double getAmount() {
    return amount;
  }
  public String getIncomeType() {
    return incomeType;
  }

}

public class StaffLoan {
    private String loanType;
    private double amount;

  public StaffLoan() {
    this( "Unknown",0.0);
  }

  public StaffLoan(String type,double amount) {
      loanType=type;
      this.amount = amount;
  }

  public double getAmount() {
    return amount;
  }
  public String getLoanType() {
    return loanType;
  }

}
```

```
import java.text.*;
public class Main {

  public static void main(String[] args) {

  DecimalFormat df = new DecimalFormat("0.00");
  Company    company = new Company();
  CompanyStaff staff;

  company.addEmployee("Syahrir","Project Manager");
  staff = company.getStaff(0);
  staff.addStaffIncome(new StaffIncome("Basic Salary",10000.00));
  staff.addStaffIncome(new StaffIncome("Bonuses",5000.00));
  staff.addStaffLoan(new StaffLoan("House Loan",2000.00));
  staff.addStaffLoan(new StaffLoan("Car Loan",2000.00));

  company.addEmployee("Adrian","Software Designer");
  staff = company.getStaff(1);
  staff.addStaffIncome(new StaffIncome("Basic Salary",8000.00));
  staff.addStaffLoan(new StaffLoan("House Loan",2000.00));
  staff.addStaffLoan(new StaffLoan("Car Loan",1000.00));

  company.addEmployee("Nabilah","Senior Programmer");
  staff = company.getStaff(2);
  staff.addStaffIncome(new StaffIncome("Basic Salary",6000.00));
  staff.addStaffIncome(new StaffIncome("Overtime Allowance",2000.00));
  staff.addStaffLoan(new StaffLoan("Car Loan",1000.00));


   // Generate a report
  System.out.println("\t\t\tSTAFF INCOME REPORT");
  System.out.println("\t\t\t================");

  for ( int staff_idx = 0; staff_idx < company.getNumOfStaff(); staff_idx++ )
  {

   staff = company.getStaff(staff_idx);

   System.out.println();
   System.out.println("Staff Name: " + staff.getName());
   System.out.println("Position: " + staff.getPosition());

   for ( int income_idx = 0; income_idx < staff.getNumOfIncome(); income_idx++ )
   {
    StaffIncome staffIncome = staff.getStaffIncome(income_idx);

             System.out.println("    " + staffIncome.getIncomeType() + ": Amount per month is $"
                    + df.format(staffIncome.getAmount()));
   }

   for ( int loan_idx = 0; loan_idx < staff.getNumOfLoan(); loan_idx++ )
   {
    StaffLoan staffLoan = staff.getStaffLoan(loan_idx);

             System.out.println("    " + staffLoan.getLoanType() + ": Loan per month is $"
                    + df.format(staffLoan.getAmount()));
   }

        System.out.println("    " + "Total Income per month is: $"
                    + df.format(staff.getTotalIncome()));

        System.out.println("    " + "Total Loan per month is: $"
                    + df.format(staff.getTotalLoan()));

        System.out.println("    " + "Net Income per month: $"
                    + df.format(staff.getNetIncome()));


  }
  }

}
```