27/10/2020

classmate Date Page

# Week 7 - INHERITANCE PRACTICE.

1 Develop a java program to create a cl Student whose variables are usn, name & se Derive a class test from Student. It include an array of cie marks of each course & their corresponding credits in another array. Derive a class Exam from Test which includes an array of see mark Derive a class Result which calculates the grade for each course & the SGPA Create n students objects & displays all the above details.

```
import java.util.Scanner;

class Student {
    String USN, name;
    int sem;
    Scanner sc = new Scanner(System.in);
    void setStuDetails() {
     System.out.println("Enter USN of Student");
     this.USN = sc.nextLine();
     System.out.println("Enter Name of Student");
     this.name = sc.nextLine();
     System.out.println("Enter Semester of
                         Student:");
     this.sem = sc.nextInt();
    }

    Void getStudDetails() {
     System.out.println("USN:" + this.USN);
     System.out.println("Name:" + this.name);
     System.out.println("Semester:" + this.sem);
    }
}
```

```
class Test extends Student{
    double cieMarks[] = new double[5];
    int credits[] = new int[5];
    int totalCredits = 0;
    void setCieDetails() {
        for(int i=0; i<cieMarks.length; i++) {
            System.out.println("Enter CIE marks(50)
                    in Course" + (i+1) + ": ");
            cieMarks[i] = sc.nextDouble();
            System.out.println("Enter credits of Course" +
                    (i+1) + ":");
            credits[i] = sc.nextInt();
            totalCredits += credits[i];
        }
    }
}

class Exam extends Test{
    double seeMarks[] = new double[5];
    double totalMarks[] = new double[5];
    int totCredits = Super.totalCredits;
    void setSeeDetails() {
        for(int i=0; i<cieMarks.length; i++){
            System.out.println("Enter SEE marks(100)
                    in course" + (i+1) + ":");
            seeMarks[i] = sc.nextDouble()/2;
        }
        calcTotalMarks();
    }
    void calcTotalMarks() {
        for(int i=0; i<5; i++) {
            totalMarks[i] = cieMarks[i] + seeMarks[i];
        }
    }
}
```

```
class Result extends Exam {
    char grades[] = new char[5];
    double sgpa = 0;
    int points[] = new int[5];
    void calcSGPA() {
        for (int i = 0; i < 5; i++) {
            if (totalMarks[i] > 100) {
                System.out.println("Error : Marks are above
                return;
            } else if (totalMarks[i] >= 90) {
                points[i] = 10;
            } else if (totalMarks[i] >= 80) {
                points[i] = 9;
            } else if (totalMarks[i] >= 70) {
                points[i] = 8;
            } else if (totalMarks[i] >= 60) {
                points[i] = 7;
            } else if (totalMarks[i] >= 50) {
                points[i] = 5;
            } else if (totalMarks[i] >= 40) {
                points[i] = 4;
            } else {
                points[i] = 0;
            }

            sgpa += (points[i] * credits[i]);
        }
    }

    void calcGrade() {
        for(int i=0; i<5; i++) {
            if (totalMarks[i] > 100) {
                System.out.println("Error: Marks are above
                return;
```

```
} else if (totalMarks[i] >= 90) {
        grades[i] = 'S';
    } else if (totalMarks[i] >= 80) {
        grades[i] = 'A';
    } else if (totalMarks[i] >= 70) {
        grades[i] = 'B';
    } else if (totalMarks[i] >= 60) {
        grades[i] = 'C';
    } else if (totalMarks[i] >= 50) {
        grades[i] = 'D';
    } else if (totalMarks[i] >= 40) {
        grades[i] = 'E';
    } else {
        points[i] = 'F';
    }
  }
}

void getSGPA() {
  System.out.format("SGPA is %.2f",
                    (sgpa / total(Credits)));
  }

void getGrades() {
  for(int i=0; i<5; i++) {
    System.out.println("Subject "+(i+1)+":"+
                       grades[i]);
  }
}

public class SGPACalculator{
  public static void main(String[] args) {
    int n=0;
```

```
Scanner sc= new Scanner(System.in);
System.out.println("Enter Number of
                    Students");
n= sc.nextInt();
Result results[] = new Result[n];
for(int i=0; i<n; i++){
    results[i]= new Result();
    results[i].setStuDetails();
    results[i].setCieDetails();
    results[i].setSeeDetails();
    results[i].calcSgpa();
    results[i].calcGrade();
    results[i].getSgpa();
    results[i].getGrades();

  }
 }
}
```

Week 7 - Inheritance

classmate
Date
Page

2) Develop a Java program to create a class Player with member variables name, matches-played & average? This class has an abstract method cal-average (String, int, int). Derive two classes BATSMAN & BOWLER from PLAYER. Class BATSMAN has a member variable runs-scored. class BOWLER has a member variable runs-given. Create m BATSMAN objects & n BOWLER object. calculat & display the average runs scored by each BATSMAN & average runs given by each BOWLER.

```
import java.util.*;

abstract class player {
    String name;
    int matchesplayed;
    double avg;
    Scanner S = new Scanner(System.in);

abstract public String getName();
abstract public void SetName();
abstract public int getMatches().
abstract public void setMatches();
abstract public double getAvgScore();
abstract public void setAvgScore();
abstract public void printData();
}

class Batsman extends player {
    private double runsScored;

public Batsman() {
```

```
    SetData();
}

@override
public String getName() {
    return this.name;
}

@override
public void setName() {
    String name = S.nextLine();
    this.name = name;
}

@override
public int getMatches() {
    return matchesplayed;
}

@Override
public void setMatches() {
    int matches = S.nextInt();
    this.matchesplayed = matches;
}
    public double getRunsScored() {
        return runsScored;
    }

public void SetRunsScored() {
    double rune = S.nextDouble();
    this.runsScored = rune;
}

public void SetData() {
```

```
System.out.println("Enter batsman name.");
SetName();
System.out.println("Enter total runs scored
  from all matches.");
SetRunsScored();
}

@override
public double getAvgScore() {
  return avg;
}

@override
public void SetAvgScore() {
  this.avg = getRunsScored() / getMatches();
}

@Override
public void printData() {
SetAvgScore();

System.out.println("Name = " + this.getName() +"
  \n Total runsScored = " + this.getRunsScored() +
  \n Average score = " + this.getAvgScore());
}
}
class Bowler extends player {
  private double runsGiven;

public Bowler() {
  setData();
}

@override
```

```
public String getName() {
  return this.name;
}

@override
public void SetName() {
  String name = S.nextLine();
  this.name = name;
}

@Overyride
public int getMatches() {
  return matchesPlayed;
}

@override
public void SetMatches() {
  int matches = S.nextInt();
  this.matchesplayed = matches;
}

@overrride
public double getAvgScore() {
  return avg;
}

Public double getRunsGiven() {
  return this.runsGiven;
}

public void SetRunsGiven() {
  double runs = S.nextDouble();
  this.runsGiven = runs;
}
```

```
public void setData() {
 System.out.println("Enter bowler name.");
 setName();
System.out.println("Enter total number of
                matches.");
 setMatches();
System.out.println("Enter total runs given
      from all matches.");
 setRunsGiven();
}

 @Override
 public void SetAvgScore() {
   this.avg = getRunsGiven() / getMatches();
 }

 @Override
  public void printData() {
   SetAvgScore();
 System.out.println("Name = " + this.getName() +
    "\n Total runs given = " + this.getRuns
   + \n Average Score = " + this.getAvgScore());
  }
}

class playerAvg {
  public static void main(String[] args) {
   int m,n, i;
  Scanner s = new Scanner(System.in);

  System.out.println("Enter number of
          batsman");
  m = s.nextInt();
  player bat[] = new Batsman[m];
```

```
System.out.println("Enter number of bowl
n = s.nextInt();
player ball[] = new Bowler[n];

for (i=0; i<m; i++){
  bat[i] = new Batsman();
  bat[i].printData();
}

  for (i=0; i<n; i++){
    ball[i] = new Bowler();
    ball[i] = printData();
  }
}
}
```