1) Class Student ← → variables → usn, name, sem.

derive Class Test from student → array of cie marks of each course and corresponding credits in another array.

Exam from test → array of see marks

Class Result → to calc grades for each course and SGPA.

N student objects and display details.

A) class Student
{

    String usn, name, sem;
    void readname()
    {
      Scanner ss = new Scanner(System.in);
      usn = ss.next();
      name = ss.next();
      sem = ss.next();
    }

class test extends students
{
    int n, i;
    double cie[];
    int credits[];
    void read marks()
    {
      System.out.println("enter no. of courses");
      Scanner ss = new Scanner(System.in);
      n = ss.nextInt();

```java
cie = new double[n];
credits = new int[n];
System.out.println("enter cie and credits ");
for(i=0; i<n; i++)
{
    cie[i] = ss.nextDouble();
    credits[i] = ss.nextInt();
}
}
}
class exam extends test
{
double see[];
Void readsee()
{
System.out.println("enter see marks");
Scanner ss = new Scanner(System.in);
See = new double[n];
for(i=0; i<n; i++)
    see[i] = ss.nextDouble();
}}
```

```java
class result extends exam
{
float result;
int score[] = new int[n+1];
void calc()
{
    int float Sum=0, sumc=0;
    for(i=0; i<n; i++)
    {
        double z = cie[i] + (see[i]/2);
        if(z >= 90)
            Score[i] = 10;
        else if(z >= 80)
            Score[i] = 9;
        :
        Sum += score[i] * credits[i];
        Sumc += credits[i];
    }
    result = Sum/sumc;
void display()
System.out.println("name = "+name+"uspe"+usn+"sem "+sem+"sgpa"+result);
}
```

```
class   sgpa
{  public static void main(String args[])
{  system.out.println ("enter no. of students");
   Scanner ss=new Scanner (System.in);
   int x=ss. nextInt();
   result res[]= new result[x];
   float y;
   for(i=0; i<x; i++)
   {
      res[i]=new result();
      System.out.println ('enter usn name sem");
      res[i] = readname();
      res[i]. readmarks();
      res[i]. readsec();
      res[i]. calc();
      System.out.println ("the details are ");
      res[i]. display();
   }
}
}
```

2) Class Player — name, matches_played, avg
   ↓ — has abstract method cal_average (String, int, int)
   derive 2
   Batsman — runs_scored
   Bowler — runs_given
   m batsman, n bowler objects. Calc and display avg
   runs scored by each Batsman and avg given by bowler.

A)
```
abstract   class   Player
{
    String name;
    int matches_played;
    double avg;
    abstract void cal_average (String s, int m, int r);
}

class Batsman   extends Player
{ int runs_scored;
  void cal_average (String s, int m, int r);
  { name =s;
    matches_played=m;
    runs_scored = r;
    avg = runs_scored/matches_played;
    System.out.println("Avg = " +avg);
}}
```

```
class Bowler extends Player
{
    int runs_given;
    void cal_average (string s, int m, int r)
    {
        name = s;
        matches_played = m;
        runs_given = r;
        avg = runs_given / matches_played;
        System.out.println ("Avg " + avg);
    }
}

class Cricket
{
    public static void main (String args[])
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter no. of bowlers");
        int bo = sc.nextInt();
        System.out.println (" no. of batsman");
        int ba = sc.nextInt();
        Bowler b1[] = new Bowler [bo];
        Batsmen b2[] = new Batsman [ba];
        System.out.println ("the bowlers are ");
        for (i=0; i<bo; i++)
        {
            b1[i] = new Bowler();
            System.out.println ("Enter name, matches, runs given");
            String boname = sc.next();
            int bomatch = sc.nextInt();
            int boruns = sc.nextInt();
            b1[i].cal_average (boname, bomatch, boruns);
        }
```

```
        System.out.println (" The batsmen are");
        for (i=0; i<ba; i++)
        {
            b2[i] = new Batsman();
            System.out.println ("Enter the name of, matches, runs scored");
            String baname = sc.next();
            int bamatch = sc.nextInt();
            int baruns = sc.nextInt();
            b2[i].cal_average (baname, bamatch, baruns);
        }
    }
}
```