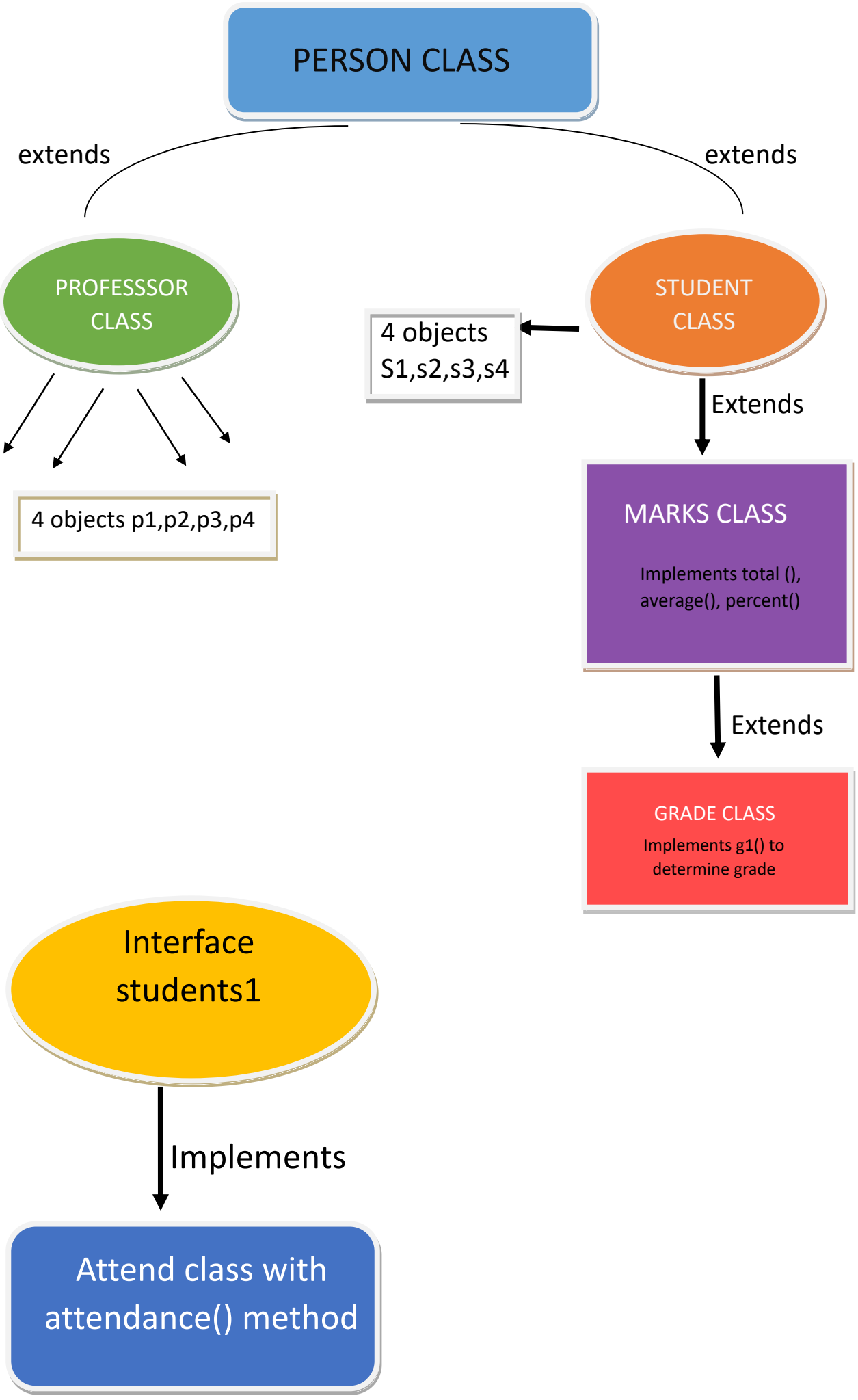
Assignment 2 (Project)
Department of Mathematical Sciences
Indian Institute of Technology (BHU)-Varanasi.
MA-104: ITCW

Problem Statement:

- Implement person class and Student Class. Using abstract classes and OOPs Concepts- inheritance, interface, abstraction create classes to perform different task. Personal Details such as Roll number, name, sex, branch, and marks in 3 subjects MA104, CSO102 and PHY102 must be passed by value (not user input).
- Calculate Total marks, average and percentage of marks using different methods. Create suitable criteria for grades and assign a grade according to the percentage of marks. Implement an interface containing a subclass to check the fulfilment of attendance criteria (75%) of each student.
- Dynamically create an array to find the student who obtained the highest marks among all and display his/her name.
- Create a subclass Professor of parent class Person to display the personal Details (Employee number, name, sex, age and Branch taught) of professors

Classes and Subclasses Used:

1. Abstract class person Class- Pre implemented
2. student Class -Pre implemented
3. professor Class (Subclass of person class to display details of professors)
4. marks class (Subclass of student class)
5. grade class (Subclass of marks class)
6. An interface students1
7. attend class (Subclass of interface Students1)
8. Public class MyTest1 containing main() method


PERSON CLASS
extends
extends
PROFESSSOR CLASS
STUDENT CLASS
4 objects S1,s2,s3,s4
Extends
4 objects p1,p2,p3,p4
MARKS CLASS
Implements total (), average(), percent()
Extends
GRADE CLASS
Implements g1() to determine grade
Interface students1
Implements
Attend class with attendance() method

Description of different functions classes perform

Abstract class person:

```
  String getName()//function to return name
  {
  return name;
  }
  void Display()//displays basic person information
  {
  System.out.println("Name = "+name);
  System.out.println("Sex = "+sex);
  System.out.println("Age = "+age);
  }
}
```

getName() returns the name passed and Display() prints the required information

Class student

```
  student(String name, char sex, int age, int RollNo, String branch)
  {
  super(name, sex, age); // calls parent class's constructor with 3 arguments
  this.RollNo = RollNo;
  this.branch = branch;
  }
  void Display() // Method Overriding
  {
  System.out.println("Roll No = "+RollNo);
  System.out.println("Name = "+getName());// calling getname() function to return name
  System.out.println("Sex = "+sex);
  System.out.println("Age = "+age);
  System.out.println("Branch = "+branch);
  }
  void TestMethod() // test what is valid to access it is not called
  {
  // name = "Mark"; Error: name is private
  sex = 'M';
  RollNo = 20;
  }
}
```

It contains a constructor of student class and Display() function of Method overriding to print personal details

Class Professor

```
professor(String name, char sex, int age, int Empno, String branch)
{
super(name, sex, age); // calls parent class's constructor with 3 arguments
this.Empno = Empno;
this.branch=branch;
}
void Display() // Method Overriding
{
System.out.println("Employee Number = "+Empno);
System.out.println("Name = "+getName());// calling getname() function to return name
System.out.println("Sex = "+sex);
System.out.println("Age = "+age);
System.out.println("Branch Taught= "+ branch);
}
void TestMethod() // test what is valid to access
{
// name = "Mark"; Error: name is private
sex = 'M';
Empno = 25;
}
```

Class professor contains a constructor to assign values and Display() function to print information of the employee along with a TestMethod() to check what is valid to access.

Class marks

```
public int MA104;
public int CSO102;
public int PHY102,sum,av,p;
void total()// method to calculate total marks
{
    sum=MA104+CSO102+PHY102;
}
void average()// method to calculate average of 3 subjects
{
    av=(MA104+CSO102+PHY102)/3;
}
void per()// method to calculate percentage of 3 subjects
{
    p=(sum*100)/300;
}
int percent()
{
    return (sum*100)/300;
}
void Display()// Method overriding
{
    System.out.println("Roll No = "+RollNo);
     System.out.println("Name = "+getName());
     System.out.println("Sex = "+sex);
     System.out.println("Age = "+age);
     System.out.println("Branch = "+branch);
     System.out.println("Total = "+sum);
     System.out.println("Average = "+av);
     System.out.println("Percentage = "+p);
```

Class mark contain void total() to calculate total marks method average(), method per(), percent() to calculate average marks and percentage of marks respectively. Display() prints all personal information.

Class grade

```
String g;
    void g1() //a method of grade class to determine grade
    {
        if (p>=90 && p<=100)
            g="A+";
        else if(p>=80 && p<90)
            g="A";
        else if(p>=70 && p<80)
            g="B+";
        else if(p>=60 && p<90)
            g="B";
        else if(p>=50 && p<60)
            g="C";
        else g="D";
    }
    void Display() //a method to display all details (METHOD OVERRIDING)
    {

        System.out.println("Roll No = "+RollNo);
         System.out.println("Name = "+getName());
         System.out.println("Sex = "+sex);
         System.out.println("Age = "+age);
         System.out.println("Branch = "+branch);
         System.out.println("Total = "+sum);
         System.out.println("Average = "+av);
         System.out.println("Percentage = "+p);
         System.out.println("Grade = "+g);

    }
```

Void g1() method checks the percentage p value and assigns a grade according to the criteria and void Display() prints all the information obtained till now.

Interface students1

```
}
interface students1 //a students interface for attendance criteria
{
    public void attendance();
}
class attend implements students1 //a class attend to implement interface
{
    private int a;
    attend(int a) //constructor of attend class
    {
        this.a=a;
    }
    public void attendance() //method of class attend for printing attendance status
    {if(a>=75)
        System.out.println("CONGRATS, ATTENDANCE ABOVE 75%");
    else System.out.println("OH NO! YOUR ATTENDANCE BELOW 75%");

    }
}
```

Interface students1 declares a public method void attendance(). A subclass attend of students1 implements it and defines it as a basis of 75% attendance.

Public class Mytest1

```
public class Mytest1
{
 public static void main(String args[] )// Main function calls all function
 {// Create object of student grade, Professor and attend class and pass values
 grade s1 = new grade("Anshika", 'F', 19, 1, "Computer Science",87,54,78);
 grade s2 = new grade("Aastha Kandwal", 'F', 19, 2, "Software Engineering",76,98,67);
 grade s3 = new grade("Debangi Ghosh", 'F', 19, 3, "Mathematics and Computing",76,98,99);
 grade s4 = new grade("Shashank Shekhar Singh", 'M', 20, 4, "Mechanical Engineering",76,99,99);
 professor p1 = new professor("Devika S Menon", 'F', 39, 55210, "Computer Science");
 professor p2 = new professor("Tanisha Jamod", 'F', 28, 87765, "Mathematics and Computing");
 professor p3 = new professor("Abhishek Kumar", 'M', 30, 94091, "Software Engineering");
 professor p4 = new professor("Harsh Sinha", 'M', 27, 93409, "Mechanical engineering");
 attend at1=new attend(76);// calling attendance interface and passing values
 attend at2=new attend(54);
 attend at3=new attend(89);
 attend at4=new attend(87);
```

Public class Mytest1 creates object of grade class and professor class to print all the details of each student and professors by passing values. Similiarly, attendance percentage is passed and it is compared with 75% in attendance method later.

```
int arr[]=new int[4];// an array to store percentage of all students
arr[0]=s1.percent();
arr[1]=s2.percent();
arr[2]=s3.percent();
arr[3]=s4.percent();
int l=0;int c=0;String n1;
for(int i=0;i<4;i++)
{
    if(arr[i]>l)
    {
        l=arr[i];// to store max marks among all
        c=i;
    }
}
if(c==0)  //n1 to store the name of the topper
    n1=s1.getName();
else if(c==1)
    n1=s2.getName();
else if(c==2)
    n1=s3.getName();
else
    n1=s4.getName();
System.out.println("TOPPER AMONG ALL THE STUDENTS");
System .out.println("Name: "+n1+"    "+"Percentage is: "+l); //Display topper name
System.out.println();
```

It then dynamically creates an array to store percentage of four students and print the name and percentage of The topper among them.

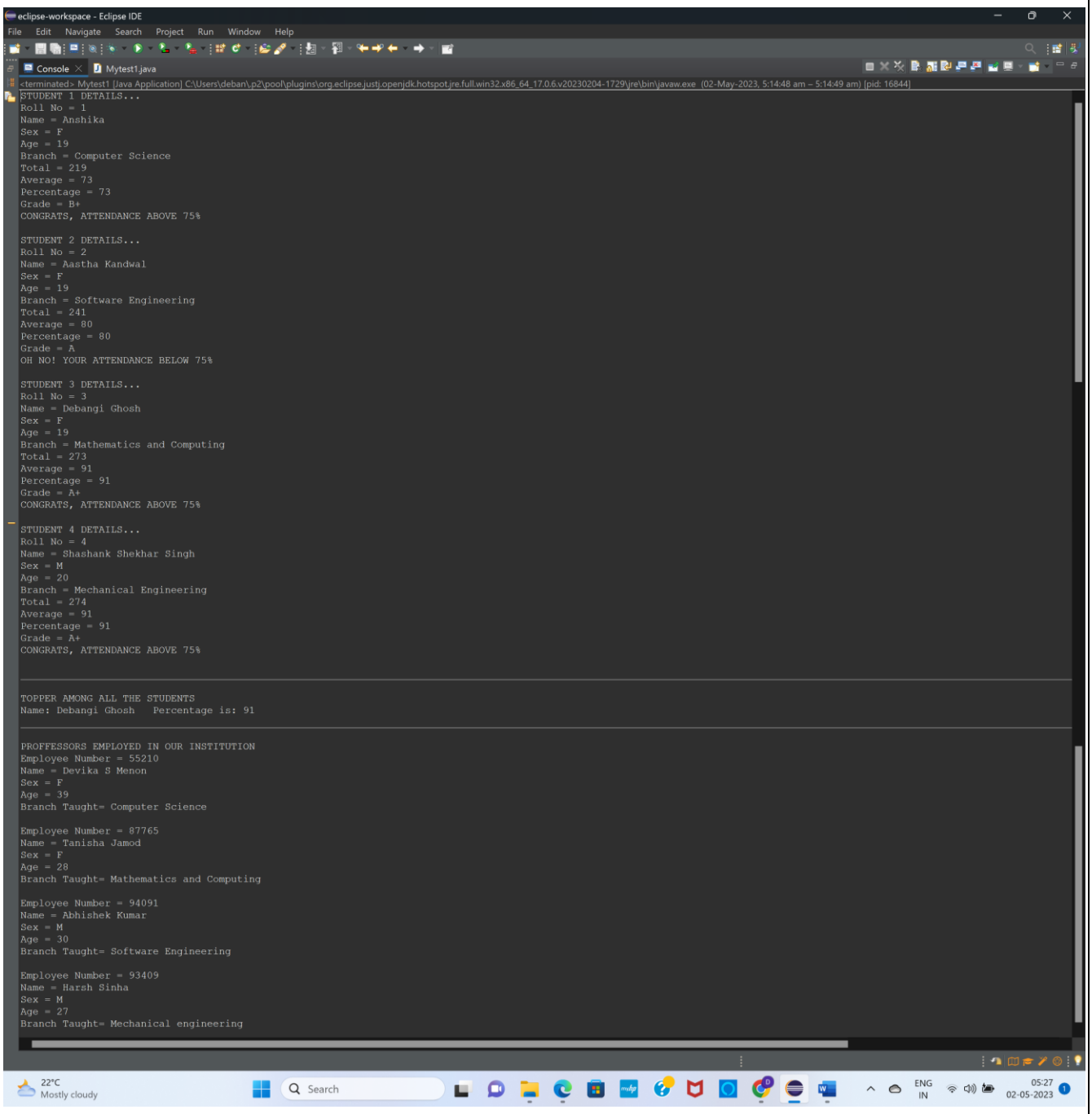
```
System.out.println("STUDENT 1 DETAILS...");
s1.total();
s1.average();
s1.per();
s1.g1();
s1.Display();
at1.attendance();
System.out.println();
```

The object created for each student calls all method of mark, attend class.

```
System.out.println();
System.out.println("PROFFESSORS EMPLOYED IN OUR INSTITUTION");// Professor Details in our institution
p1.Display();
System.out.println();
p2.Display();
System.out.println();
p3.Display();
System.out.println();
p4.Display();
 }
}
```

After inserting a line, employee details of all professors are printed using an object of poofessor class.

Code Output-



Created by-
Debangi Ghosh (22124013)
Anshika Yadav (22124010)
Branch- Mathematics and Computing