Student

Java

```java
//This program generates a daily schedule for each Student. If the Student is just a Student type, they get a six period schedule but have the option to add in other extracurriculars they do after school. Athletes have a similar schedule but instead of PE, they have their sport practice after their classes. Swimmers have a swim meet in their schedule after they have practice. There are additional methods that allow the Student to edit their schedule and compare their schedule to the other Students
//UML Diagram - https://docs.google.com/drawings/d/1_Jw3DMX-uFNutGtjbBVLW2KkHilRnVR_4xRoYAZ9p60/edit?usp=sharing

import java.util.ArrayList;
public class Main {

   public static void main(String[] args) {
       //can the non overriden method called be a mutator/accessor method?
       Student johnny = new Student("Johnny", 11);
       Student ricky = new Athlete("Ricky", 10, 6, "Basketball");
       Student anton = new Swimmer("Anton", 10, 2, 39.6);
       Athlete s1 = new Athlete("Mark", 11, 2, "Basketball");

       //making first swimmer time array
       ArrayList<Double> times1 = new ArrayList<Double>();
       times1.add(39.03);
       times1.add(40.5);
       times1.add(36.78);
       times1.add(32.65);
       Athlete s2 = new Swimmer("Terry", 11, 8, times1);
       //making second swimmer time array
       ArrayList<Double> times2 = new ArrayList<Double>();
       times2.add(40.63);
```

```
        times2.add(43.65);
        times2.add(34.53);
        times2.add(29.32);
        Swimmer s3 = new Swimmer("Kai", 11, 5, times2, 29.32);

        //generating schedule for students
        ((Athlete)ricky).generateSchedule(); //casting
        ((Swimmer)anton).generateSchedule();
        johnny.generateSchedule();
        s1.generateSchedule();
        ((Swimmer)s2).generateSchedule(); //casting
        s3.generateSchedule();

        ricky.addExtraCurricular(); //add extracurricular
        s3.addExtraCurricular(); //add extracurricular
        johnny.removeClass("English"); //remove class
        System.out.println();
        System.out.println("Most number of classes: " +
Student.mostBusyStudent()); //number of classes that student with most classes
has

        //toString
        System.out.println();
        System.out.println(s1);
        System.out.println();
        System.out.println(anton);

        //print schedule
        System.out.println();
        System.out.println(ricky.getName() + "'s' schedule: ");
        ricky.printSchedule();
        System.out.println();
        System.out.println(s1.getName() + "'s schedule: ");
        s1.printSchedule();
```

```
        System.out.println();
        //after school
        ((Athlete)ricky).afterSchool();
        System.out.println();
        johnny.afterSchool();
        //get best time
        System.out.println();
        ((Swimmer)s2).calculateBestTime();
        System.out.println(s2);
        System.out.println();

        //equals
        System.out.println("Do Ricky and Mark do the same sport?: " +
((Athlete)ricky).equals(s1));
        System.out.println("Do Terry and Kai have the same best time?: " +
s2.equals(s3));

        //display times
        System.out.println();
        s3.displayTimes();
        System.out.println();

        //seeing who has more experience
        System.out.println("Athlete with more experience: " +
((Athlete)ricky).moreExperience(s3));
        System.out.println("Athlete with more experience: " +
((Athlete)anton).moreExperience(s1));  //chooses random because they have the
same experience.

    }
}
```

Athlete

Java

```java
import java.util.*;
public class Athlete extends Student{
   private int years;
   private String sportName;

   //CONSTRUCTOR
   public Athlete(String name, int grade, int years, String sportName){
       super(name, grade);
       this.years = years;
       this.sportName = sportName;
   }
   //ACCESSOR
   public int getYears(){
       return years;
   }
   public String getSportName(){
       return sportName;
   }
   //MUTATOR
   public void setYears(int years){
       this.years = years;
   }
   public void setSportName(String sportName){
       this.sportName = sportName;
   }
   //OVERRIDEN METHODS
   public void generateSchedule(){ //takes out pe and adds sport practice after
school
       super.generateSchedule();
```

```
        int num = -1;
        ArrayList<Class> temp = this.getSchedule();
        for(int i = 0; i < temp.size(); i++){
            if(temp.get(i).getClassName().equals("PE")){
                temp.remove(i);
                num = i;
                int period = temp.size()+1;
                Class sport = new Class(sportName, period);
                temp.add(sport);
            }
        }
        for(int i = 0; i < temp.size(); i++){
            temp.get(i).setPeriod(i+1);
        }
        this.setSchedule(temp);
    }
    public void afterSchool(){ //print statement overriden method
        System.out.println(this.getName() + ", it is time to go to practice");
    }

    //OTHER METHODS
    public boolean equals(Object obj){ //compares if athletes have the same
sport
        Athlete a = (Athlete)obj;
        if(this.getSportName().equals(a.getSportName())){
            return true;
        }
        return false;
    }

    public Athlete moreExperience(Athlete a){ //compares which athlete has more
years of experience
        if(this.getYears() > a.getYears()){
```

```
            return this;
        }
        else if(this.getYears() < a.getYears()){
            return a;
        }
        else if(this.getYears() == a.getYears()){ //if tie picks a random
athlete
            int choice = (int)(Math.random() * 2);
            if(choice == 0)
                return this;
            else if(choice == 1)
                return a;
        }
        return this;
    }

    //toString OVERRIDEN
    public String toString(){
        return super.toString() + ", Sport name: " + sportName + ", Years doing
sport: " + years;
    }
}
```

Swimmer

Java

```java
import java.util.ArrayList;
public class Swimmer extends Athlete {
   private ArrayList<Double> times;
   private double bestTime;

   //constructors
   public Swimmer(String name, int grade, int years, double bestTime){
       super(name, grade, years, "Swimmming");
       this.bestTime = bestTime;
       times = new ArrayList<Double>();
   }
   public Swimmer(String name, int grade, int years, ArrayList<Double> times){
       super(name, grade, years, "Swimming");
       bestTime = 0.0;
       this.times = times;
   }
   public Swimmer(String name, int grade, int years, ArrayList<Double> times,
double bestTime){
       super(name, grade, years, "Swimming");
       this.bestTime = bestTime;
       this.times = times;
   }

   //ACCESSOR
   public double getBestTime(){
       return bestTime;
   }
   public ArrayList<Double> getTimes(){
       return times;
   }
   //MUTATOR
   public void setBestTime(double bestTime){
       this.bestTime = bestTime;
```

```
    }
    public void setTimes(ArrayList<Double> times){
        this.times = times;
    }

    public void addTime(double time){ //adds time to arrayList
        times.add(time);
    }

    //OVERRIDEN METHOD
    public void generateSchedule(){ //adds swim tournament to student schedule
        super.generateSchedule();
        String swimMeet = "Swim tournament";
        int period = this.getSchedule().size() + 1;
        Class meet = new Class(swimMeet, period);
        this.getSchedule().add(meet);
    }

    public void afterSchool(){ //overriden print statement
        System.out.println(this.getName() + ", it is time for a swim practice
and a swim tournament");
    }

    public double calculateBestTime(){ //swimmer can find their best time
        double fastestTime = 120940.6;
        for(int i = 0; i < times.size(); i++)
        {
            if(times.get(i) < fastestTime){
                fastestTime = times.get(i);
            }
        }
        bestTime = fastestTime;
        return fastestTime;
    }
```

```
   //OTHER METHODS
   public void displayTimes(){ //displays all swimmers times in times array - similar to accessor but in a more organized format
        System.out.println(this.getName() + "'s times");
        for(int i = 0; i < times.size(); i++){
            System.out.println("Time: " + times.get(i));
        }
   }

   public boolean equals(Object obj){ //overriding equals method - if swimmers have the same best time and are in the same grade
        Swimmer swim = (Swimmer)obj;
        if(this.bestTime == swim.bestTime && this.getGrade() == swim.getGrade()){
            return true;
        }
        return false;
   }
   //toString OVERRIDEN
   public String toString(){
        return super.toString() + ", Swimmers Best Time: " + bestTime;
   }
}
```

Class

Java

```java
public class Class{  //object used to define a class (event) in student's
schedule
  private String className;
  private int period;
   //constructors
  public Class(String name){
      className = name;
      period = 0;
  }
  public Class(String name, int period){
      className = name;
      this.period = period;
  }
  //mutator
  public void setClassName(String name){
      className = name;
  }
  public void setPeriod(int period){
      this.period = period;
  }
  //accessor
  public String getClassName(){
      return className;
  }
  public int getPeriod(){
      return period;
  }
  //toString
  public String toString(){
      return "Hour " + period + ", " + className;
  }
}
```

Main:

Java

```java
//Ilina Iyer - This program generates a daily schedule for each Student. If the Student is just a Student type, they get a six period schedule but have the option to add in other extracurriculars they do after school. Athletes have a similar schedule but instead of PE, they have their sport practice after their classes. Swimmers have a swim meet in their schedule after they have practice. There are additional methods that allow the Student to edit their schedule and compare their schedule to the other Students
//UML Diagram - https://docs.google.com/drawings/d/1_Jw3DMX-uFNutGtjbBVLW2KkHilRnVR_4xRoYAZ9p60/edit?usp=sharing

import java.util.ArrayList;
public class Main {

   public static void main(String[] args) {
       //can the non overriden method called be a mutator/accessor method?
       Student johnny = new Student("Johnny", 11);
       Student ricky = new Athlete("Ricky", 10, 6, "Basketball");
       Student anton = new Swimmer("Anton", 10, 2, 39.6);
       Athlete s1 = new Athlete("Mark", 11, 2, "Basketball");

       //making first swimmer time array
       ArrayList<Double> times1 = new ArrayList<Double>();
       times1.add(39.03);
       times1.add(40.5);
       times1.add(36.78);
       times1.add(32.65);
       Athlete s2 = new Swimmer("Terry", 11, 8, times1);
       //making second swimmer time array
       ArrayList<Double> times2 = new ArrayList<Double>();
```

```
        times2.add(40.63);
        times2.add(43.65);
        times2.add(34.53);
        times2.add(29.32);
        Swimmer s3 = new Swimmer("Kai", 11, 5, times2, 29.32);

        //generating schedule for students
        ((Athlete)ricky).generateSchedule(); //casting
        ((Swimmer)anton).generateSchedule();
        johnny.generateSchedule();
        s1.generateSchedule();
        ((Swimmer)s2).generateSchedule(); //casting
        s3.generateSchedule();

        ricky.addExtraCurricular(); //add extracurricular
        s3.addExtraCurricular(); //add extracurricular
        johnny.removeClass("English"); //remove class
        System.out.println();
        System.out.println("Most number of classes: " +
Student.mostBusyStudent()); //number of classes that student with most classes
has

        //toString
        System.out.println();
        System.out.println(s1);
        System.out.println();
        System.out.println(anton);

        //print schedule
        System.out.println();
        System.out.println(ricky.getName() + "'s' schedule: ");
        ricky.printSchedule();
        System.out.println();
        System.out.println(s1.getName() + "'s schedule: ");
```

```
        s1.printSchedule();

        System.out.println();
        //after school
        ((Athlete)ricky).afterSchool();
        System.out.println();
        johnny.afterSchool();
        //get best time
        System.out.println();
        ((Swimmer)s2).calculateBestTime();
        System.out.println(s2);
        System.out.println();

        //equals
        System.out.println("Do Ricky and Mark do the same sport?: " +
((Athlete)ricky).equals(s1));
        System.out.println("Do Terry and Kai have the same best time?: " +
s2.equals(s3));

        //display times
        System.out.println();
        s3.displayTimes();
        System.out.println();

        //seeing who has more experience
        System.out.println("Athlete with more experience: " +
((Athlete)ricky).moreExperience(s3));
        System.out.println("Athlete with more experience: " +
((Athlete)anton).moreExperience(s1));  //chooses random because they have the
same experience.

    }
}
```