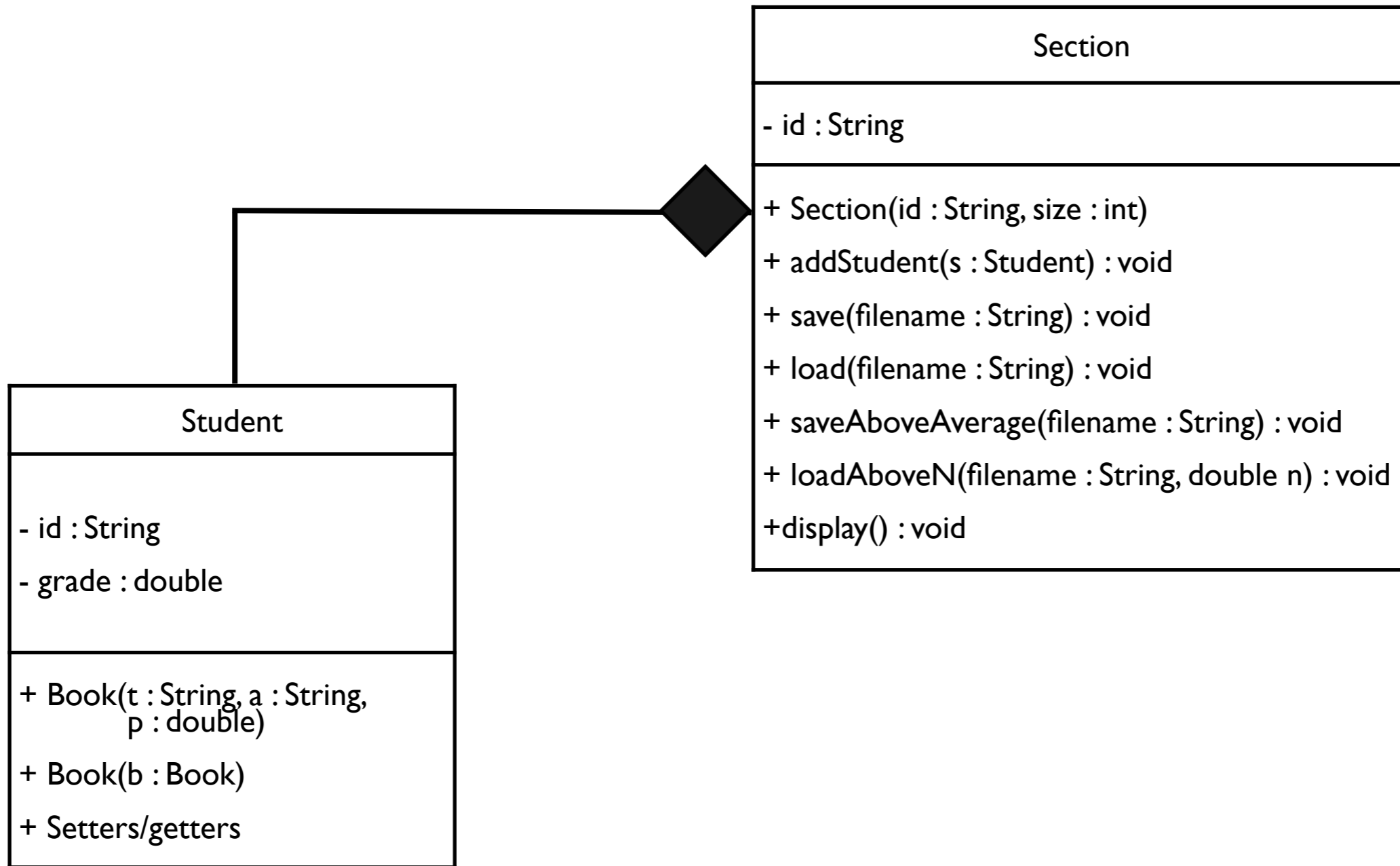


CSC 113

Tutorial 10

Object files



Class Student

```
public class Student implements Serializable{  
    String id;  
    double grade;  
  
    public Student(String id, double grade)  
    { this.id = id; this.grade = grade; }  
  
    public Student(Student s)  
    { id = s.id; grade = s.grade;}
```

Student
- id : String - grade : double
+ Book(t : String, a : String, p : double) + Book(b : Book) + Setters/getters

Class Section

```
public class Section {
    String id;
    Student[] students;
    int currentSize;

    public Section(String id, int size)
        throws NegativeArraySizeException
    {
        this.id = id;
        students = new Student[size];
        currentSize = 0;
    }

    public void addStudent(Student s)
        throws ArrayIndexOutOfBoundsException
    {
        students[currentSize]=new Student(s);
        currentSize++;
    }
}
```

Section
- id : String
+ Section(id : String, size : int) + addStudent(s : Student) : void + save(filename : String) : void + load(filename : String) : void + saveAboveAverage(filename : String) : void + loadAboveN(filename : String, double n) : void +display() : void

Class Section

Save

```
public void save(String filename)
    throws IOException
{
    File outFile = new File(filename);
    FileOutputStream outputStream = new FileOutputStream(outFile);
    ObjectOutputStream outObjectStream = new ObjectOutputStream(outputStream);

    outObjectStream.writeInt(currentSize);
    for(int i=0; i< currentSize;i++)
        outObjectStream.writeObject(students[i]);

    outObjectStream.close();
}
```

Section
- id : String
+ Section(id : String, size : int)
+ addStudent(s : Student) : void
+ save(filename : String) : void
+ load(filename : String) : void
+ saveAboveAverage(filename : String) : void
+ loadAboveN(filename : String, double n) : void
+display() : void

```
outObjectStream.writeInt(currentSize)
outObjectStream.writeObject(students)
```

Class Section

Load

```
public void load(String filename)
    throws IOException, ClassNotFoundException
{
    File inFile = new File(filename);
    FileInputStream inputStream = new FileInputStream(inFile);
    ObjectInputStream inObjectStream = new ObjectInputStream(inputStream);

    currentSize = inObjectStream.readInt();
    for(int i=0;i<currentSize; i++)
        students[i] = (Student)inObjectStream.readObject();

    inObjectStream.close();
}
```

Section
- id : String
+ Section(id : String, size : int)
+ addStudent(s : Student) : void
+ save(filename : String) : void
+ load(filename : String) : void
+ saveAboveAverage(filename : String) : void
+ loadAboveN(filename : String, double n) : void
+display() : void

```
currentSize = inObjectStream.readInt();
students = (Student[])inObjectStream.re
```

Class Section

SaveAboveAverage

```
public void saveAboveAverage(String filename)
    throws IOException
{
    File outFile = new File(filename);
    FileOutputStream outputStream = new FileOutputStream(outFile);
    ObjectOutputStream outObjectStream = new ObjectOutputStream(outputStream);
    double avg, sum=0;

    for(int i=0; i< currentSize; i++)
        sum += students[i].getGrade();

    avg = sum / currentSize;

    int count=0;
    for(int i=0; i<currentSize; i++)
        if(students[i].getGrade() >= avg)
            count++;

    outObjectStream.writeInt(count);
    for(int i=0; i< currentSize; i++)
        if(students[i].getGrade() > avg)
            outObjectStream.writeObject(students[i]);

    outObjectStream.close();
}
```

Section
- id : String
+ Section(id : String, size : int) + addStudent(s : Student) : void + save(filename : String) : void + load(filename : String) : void + saveAboveAverage(filename : String) : void + loadAboveN(filename : String, double n) : void + display() : void

Class Section

loadAboveN

```
public void loadAboveN(String filename, double n)
    throws IOException, ClassNotFoundException
{
    File inFile = new File(filename);
    FileInputStream inputStream = new FileInputStream(inFile);
    ObjectInputStream inObjectStream = new ObjectInputStream(inputStream);

    Student s;
    int size = inObjectStream.readInt();
    currentSize = 0; //reset the array
    for(int i=0;i<size; i++)
    {
        s = (Student)inObjectStream.readObject();
        if(s.getGrade() > n)
            addStudent(s);
    }
    inObjectStream.close();
}
```


Section
- id : String
+ Section(id : String, size : int) + addStudent(s : Student) : void + save(filename : String) : void + load(filename : String) : void + saveAboveAverage(filename : String) : void + loadAboveN(filename : String, double n) : void +display() : void

Main

```
Section csc113; Scanner s = new Scanner(System.in);
    System.out.print("Enter Section id: ");
String id = s.next();
    System.out.print("Enter Section size: ");
int size = s.nextInt();

    try{
        csc113 = new Section(id, size);
    }
    catch(NegativeArraySizeException e)
    {
        System.out.println("Exception: Array size should be positive");
        return;
    }
}
```

Main

```
do{
    try{
        displayMenu();
        choice = s.nextInt();
        ...
    }  Next slide...
    catch(IOException e)
    {
        System.out.println(e.getMessage());
    }
    catch(ArrayIndexOutOfBoundsException e)
    {
        System.out.println(e.getMessage());
        System.out.println("Exception: Array is full");
    }
    catch (ClassNotFoundException e)
    {
        System.out.println(e.getMessage());
    }
}while(choice != 7);
}
```

```
        switch(choice)
    {
    case 1: //addStudent
        System.out.println("Enter id: "); title = s.next();
        System.out.println("Enter grade: "); grade = s.nextDouble();
        csc113.addStudent(new Student(id, grade));
        break;
    case 2: //display
        csc113.display();
        break;
    case 3://save
        System.out.print("Enter save file name: "); csc113.save2(s.next());
        break;
    case 4://load
        System.out.print("Enter load file name: "); csc113.load2(s.next());
        break;
    case 5://save above avarage
        System.out.print("Enter Save above average file name: "); csc113.saveAboveAverage(s.next());
        break;
    case 6://load above n
        System.out.print("Enter load above n file name: "); file = s.next();
        System.out.print("Enter n: ");
        csc113.loadAboveN(file, s.nextDouble());
        break;
    case 7:
        System.out.println("Bye!");
        break;
    default:
        System.out.println("Wrong input!");
        break;
    }
```

Main