**StudentInfoClient**

Read / Write info to disk file - Kiran

Generate unique Student IDs / Verify records are unique - Rahul

Display the info using SWT - Antonio

Code to add / delete Student Records - Robin

Code to Modify student records - Calvin

Interface to link add / delete / modify code to SWT widgets - Daniel

**Student Component Presentation**

Create a 4 slide presentation using the following template:

1. Slide – Java Description of your component
   * Class Name
   * Internal (Private) Constants & Variables
   * Class Constructor & Initialization Parameters
   * Class Methods
2. Slide – Verbal Description of your component
   * Explain what your component does (Jot Notes)
   * Explain how your component works (Jot Notes)
3. Slide – Integration Requirements & Expectations
   * For Each Other Component Used By Your Component…
     1. How you will access the Object
     2. What Class Methods will you use
     3. What other expectations do you have
4. Slide – Integration Requirements & Expectations (Cont’d)
   * Same as Slide 3

**Sample Code**

**public** **class** StudentInfoClient {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

StudentDatabase studentDB = **new** StudentDatabase();

StudentRecord newStudent = **new** StudentRecord("Daniel", "Gopal", "1234");

studentDB.addStudent(newStudent);

newStudent = **new** StudentRecord("Kiran", "Hart", "5678");

studentDB.addStudent(newStudent);

newStudent = **new** StudentRecord("Antonio", "N-J", "2468");

studentDB.addStudent(newStudent);

newStudent = **new** StudentRecord("Robin", "Saran", "1357");

studentDB.addStudent(newStudent);

newStudent = **new** StudentRecord("Rahul", "Tailor", "7899");

studentDB.addStudent(newStudent);

newStudent = **new** StudentRecord("Calvin", "Ye", "1111");

studentDB.addStudent(newStudent);

System.*out*.println("Student DAtabase Printout");

System.*out*.println("=========================");

studentDB.printStudents();

}

}

**public** **class** StudentDatabase {

// The student database uses an Array of Student Records

**private** **int** index;

**private** StudentRecord [] students = **new** StudentRecord[7];

// A constructor method is used when using the class

// to create a student database object

**public** StudentDatabase() {

index = 0;

}

// The class defines methods to access records in the database

**public** **void** addStudent(StudentRecord student ) {

**this**.students[index] = student;

index++;

}

**public** StudentRecord getStudent(**int** index) {

**return** **this**.students[index];

}

**public** **void** printStudents() {

**this**.students[0].printRecord();

**this**.students[1].printRecord();

**this**.students[2].printRecord();

**this**.students[3].printRecord();

**this**.students[4].printRecord();

**this**.students[5].printRecord();

}

}

**public** **class** StudentRecord {

// A student has the following data fields

**private** String firstName;

**private** String lastName;

**private** String idNumber;

// A constructor method is used when using the class

// to create student record objects

**public** StudentRecord(String firstName, String lastName, String idNumber) {

**this**.firstName = firstName;

**this**.lastName = lastName;

**this**.idNumber = idNumber;

}

// A class defines methods to access the private data fields

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;;

}

**public** **void** setIdNumber(String idNumber) {

**this**.idNumber = idNumber;

}

**public** String getFirstName() {

**return** **this**.firstName;

}

**public** String getLastName() {

**return** **this**.lastName;

}

**public** String getIdNumber() {

**return** **this**.idNumber;

}

**public** **void** printRecord() {

System.*out*.println("First: " + firstName + ", Last: " + lastName + ", ID: " + idNumber);

}

}