File Conversion Program

Using Common Object Class

You have the following data in an Excel worksheet that needs to be exported and converted.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data Type** | **Number Decimals** | **Length** | **Notes** |
| Student ID Number | Integer | 0 | 7 |  |
| Name | String | N/A | 30 |  |
| Major | String | 0 | 30 |  |
| Grade Point Average | Float | 2 | 6 |  |
| Number Credits Taken | Float | 2 | 6 |  |
| Number Credits Earned | Float | 2 | 6 |  |
| College Year | Character | N/A | 1 | 0 = Admitted 1= Freshman  2=Sophomore  3=Junior  4=Senior |
| Is On Financial AId | Boolean | N/A |  | true – on financial aid  false – not on financial aid |

The file will be exported as a comma separated values text file and converted to a binary file. The text file (CSV) file name is student.csv and the binary file name is student.dat. Display the student's id number, name, major, and gpa on the monitor.

Planning Document – Student Class

**Program Outline:** < What is your program supposed to do? >

Create instance fields

Create Accessor/Mutator Methods

Create Display method

Create Parse method

**Methods:** < This is a list of methods you will define in your program. >

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Access Modifier(1)** | **Method Name** | **Parameters (dataType identifier)** | **Return Type (2)** |
| Constructor | public | Student() |  | object |
| Constructor | public | Student() | string inputLine | object |
| Accessor Method | public | getStudentID() |  | integer |
| Accessor Method | public | getName() |  | string |
| Accessor Method | public | getMajor() |  | string |
| Accessor Method | public | getGpa() |  | float |
| Accessor Method | public | getCreditsTaken() |  | float |
| Accessor Method | public | getCreditsEarned() |  | float |
| Accessor Method | public | getCollegeYear() |  | character |
| Accessor Method | public | getiIFinancialAid() |  | boolean |
| Mutator Method | public | setStudentID() | integer value | void |
| Mutator Method | public | setName() | string value | void |
| Mutator Method | public | setMajor() | string value | void |
| Mutator Method | public | setGpa() | float value | void |
| Mutator Method | public | setCreditsTaken() | float value | void |
| Mutator Method | public | setCreditsEarned() | float value | void |
| Mutator Method | public | setCollegeYear() | character value | void |
| Mutator Method | public | setiIFinancialAid() | boolean value | void |
| Displays all instance variables | public | displayData() |  | void |
| Populates input variables from a string | public | parse() | string inputLine | boolean |

1. Access Modifier: local, public, private, protected

2. Return Type: void, string, char, byte, short, integer, long, double, float, boolean, object, etc.

**Data Items:** < This is a list of fields (variables, constants, and objects you will need >

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Item** | **Source(1)** | **Access (2)** | **Type (3)** | **Identifier** | **Notes** |
| Student ID Number | instance | private | integer | studentID |  |
| Name | instance | private | string | name |  |
| Major | instance | private | string | major |  |
| Grade Point Average | instance | private | float | gpa |  |
| Number Credits Taken | instance | private | float | creditsTaken |  |
| Number Credits Earned | instance | private | float | creditsEarned |  |
| College Year | instance | private | character | collegeYear |  |
| Is On Financial AId | instance | private | boolean | isFinancialAid |  |
| Input Line | instance | local | string | inputLine | parse method |
| succeeded | instance | local | boolean | succeeded | parse method |

1. Source (where the data comes from): calculated, input, constant, parameter, instance, object

2. Access Modifier: local, public, private, protected

3. Data Type: string, char, byte, short, integer, long, double, float, boolean, object, etc.

**Sample Output:** < What will the user see? >

Student ID: 999999999

Name: X----------------------------X

Major: X----------------------------X

Grade Point Average: 99.99

Number Credits Taken: 999.99

Number Credits Earned: 999.99

College Year: X

Is On Financial Aid: X----X

**Test Data:** < How will you prove your program works? >

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | Value 1 | Value 2 | Value 3 | Value 4 |
| studentID | 12345 |  |  |  |
| name | Gary Smith |  |  |  |
| major | CIS |  |  |  |
| gpa | 3.85 |  |  |  |
| creditsTaken | 52 |  |  |  |
| creditsEarned | 51 |  |  |  |
| collegeYear | 2 |  |  |  |
| isFinancialAid | false |  |  |  |

Note: You made more or fewer test cases depending on your application.

**Process Design:** < What is the solution (pseudocode, IPO diagram, flowchart) >

public class Student

// Instance variables.

private integer studentID

private string name

private string major

private float gpa

private float creditsTaken

private float creditsEarned

private character collegeYear

private boolean isFinancialAid

// Constructors.

public Student() // Default constructor

return

public Student(string inputLine) // Constructor for reading and parsing an input line

parse(inputLine)

return

// Accessor/Mutator Methods

public integer getStudentID()

return studentID

public void setStudentID(integer value)

studentID = value

return

public string getName()

return name

public void setName(string value)

name = value

return

public string getMajor()

return major

public void setMajor(string value)

major = value

return

public float getGPA()

return gpa

public void setGPA(float value)

gpa = value

return

public float getCreditsTaken()

return creditsTaken

public void setCreditsTaken(float value)

craditsTaken = value

retirm

public float getCreditsEarned()

return craditsEarned

public void setCreditsEarned(float value)

creditsEarned = value

return

public charcter getCollegeYear()

return collegeYear

public void setCollegeYear(charcter value)

collegeYear = value

return

public boolean getIsFinancialAid()

return isFinancialAid

public void setIsFinancialAid(boolean value)

isFinancialAid = value

return

// DisplayData method. Displays all instance variables.

public void displayData()

display "Student ID: " + getStudentID()

display "Name: " + getName()

display "Major: + getMajor()

display "Grade Point Average: " + getGPA()

display "Number Credits Taken: " + getCreditsTaken()

display "Number Credits Earned: " + getCreditsEarned()

display "College Year: " + getCollegeYear()

display "Is Financial Aid: " + getIsFinancialAid()

return

// Parse method. Takes an input string and populates the instance variables.

public boolean parse(string inputLine)

boolean succeeded = false

string[] splitFields

splitFields = inputLine.split(",")

try

studentID = Convert.toInteger(splitFIelds[0])

name = splitFIelds[1]

major = splitFIelds[2]

gpa = Convert.toFloat(splitFIelds[3])

creditsTaken = Convert.toFloat(splitFIelds[4])

creditsEarned = Convert.toFloat(splitFIelds[5])

collegeYear = Convert.toCharacter(splitFields[6])

isFinancialAid = Convert.toBoolean(splitFIelds[7])

succeeded = true

end try

catch (Exception err)

err.getMessage()

display "Error parsing input record." + inputLine

succeeded = false

end catch

return succeeded

endClass

Planning Document - Convert Student

**Program Outline:** < This is an outline of what your program is to do. Be detailed. >

Open files

Read and parse input record

write binary record

display student id, name, major and gpa on monitor

close files

**Methods:** < This is a list of methods you will define in your program.. >

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Access Modifier(1)** | **Method Name** | **Parameters (dataType identifier)** | **Return Type (2)** |
| Main method: | public static | main() |  | void |
| Startup method | public | startup() |  | void |
| Processing method | public | processing() |  | void |
| Shutdown method | public | shutdown() |  | void |
| Read a record | public | readRecord() |  | void |
| Writes a record | public | writeRecord() |  | Boolean |
| Displays headings | public | displayHeadings() |  | void |
| Display the record | public | displayRecord() |  | void |

1. Access Modifier: local, public, private, protecteds

2. Return Type: void, string, char, byte, short, integer, long, double, float, boolean, object, etc.

**Data Items:** < This is a list of fields (variables, constants, and objects you will need. >

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Item** | **Access (1)** | **Type (2)** | **Identifier** | **Notes** |
| Student class | private | Student | student |  |
| Stream Reader | private | StreamReader | reader |  |
| Binary Writer | private | BinaryWriter | writer |  |
| Input File Name | private | string | inputFileName |  |
| Output File Name | private | string | outputFileName |  |
| End of File | private | boolean | isEOF |  |
| Input Line | local | string | inputLine |  |
| Spit Fields Array | local | string | splitFields[] |  |

1. Access: public, private, protected

2. Type: string, char, byte, short, integer, long, double, float, etc.

**Sample Output:** < What will the user see? >

Student ID Name Major GPA

99999999 X-----------------------------------X X-----------------------------------X 9.99

**Test Data:** < How will you prove your program works? >

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identifier** | Value 1 | Value 2 | Value 3 | Value 4 |
| studentID | 12345 |  |  |  |
| name | Gary Smith |  |  |  |
| major | CIS |  |  |  |
| gpa | 3.85 |  |  |  |
| creditsTaken | 52 |  |  |  |
| creditsEarned | 51 |  |  |  |
| collegeYear | 2 |  |  |  |
| isFinancialAid | false |  |  |  |

Note: You made more or fewer test cases depending on your application.

**Process Design:** < What is the solution (pseudocode, IPO diagram, flowchart)? >

public class ConvertStudent

// Class variables.

private Student student

private StreamReader reader

private BinaryWriter writer

private string inputFileName = "students.csv"

private string outputFileName = "students.dat"

private Boolean isEOF

// Main method.

public void main()

startup()

while Not isEOF

processing()

end while

shutdown()

return

// Startup method.

public void startup()

// Opens the input file

if (File.exists(inputFileName)) // Makes sure it exists.

try

reader = new StreanReader(inputFileName)

end try

catch (FileIOException err)

display err.getMessage()

display "Terminating program."

shutdown()

end catch

else

display "Missing input file"

display "Terminating program."

shutdown()

endif

// Opens the output file. Make sure it does not exist.

if (Not File.exists(outputFileName))

try

writer = new BinaryWriter(outputFileName)

end try

catch (FileIOException err)

catch (FileIOException err)

display err.getMessage()

display "Terminating program."

shutdown()

end catch

else

display "Output file already exists"

display "Terminating program."

shutdown()

endif

// Display report heading

display heading()

// Read the first record (priming read)

reader.readRecord()

return

// Processing method.

public void processing()

writeRecord()

displayRecord()

readRecord()

return

// Shutdown method.

public void shutdown()

if (Not reader==null)

reader.close()

endif

if (Not writer==null)

writer.close()

endif

return

// ReadRecord method. Read and parses the input string.

public void readRecord()

string inputLine = reader.readLine()

if (inputLine==null)

isEOF = true

else

student = new Student(inputLine)

endif

return

// WriteRecord method. Writes data in binary.

public boolean writeRecord()

Boolean isSuccessful = false

try

writer.writeInteger(student.getStudentID())

writer.writeString(student.getName())

writer.writeString(student.getMajor())

writer.writeFloat(student.getGPA())

writer.writeFloat(student.getCreditsTaken())

writer.writeFloat(student.getCreditsEarned())

writer.writeChacter(student.getCollegeYear())

writer.writeBoolean(student.getIsFinancialAid())

isSuccessful = true

end try

catch (Exception err)

display err.getMessage()

display "Error writing output record. Program terminated."

shutdown()

end catch

return

// Display Heading method.

public void displayHeading()

display " Student ID Name Major GPA"

return

// Display Record method.

public void displayRecord()

display student.getStudentID() + " " + student.getName() + " " + student.getMajor() + " " +

student.getGPA()

return

endClass