### Function: validateFile

START function: void validateFile(string filePath)

READ file at filePath

CREATE file parser

FOR all rows, r, in file

GET row size

IF row size is less than 2

OUTPUT course name and “ missing course name.”

ELSE IF row size is greater than 2

FOR each prerequisite, p

SET c to first course in file

WHILE c is not null

IF p’s ID matches c’s ID

BREAK from while loop

ELSE

SET c to next course

END IF

ELSE

OUTPUT “Prerequisite, p, not found in course catalog”

END WHILE

END FOR

END IF

END FOR

END function

### Function createCourse

START function createCourse(filePath)

CREATE file parser variable, fileParse

FOR each row, r in file

CREATE new course object, newCourse with key at hash value of the bidID

SET rowSize variable equal to length of s

SET newCourse’s course ID equal to item 1 in r

SET newCourse’s course ID equal to item 2 in r

IF s > 2

FOR each item from positions 2 to s in r

APPEND item to newCourse’s prerequisite vector

END FOR

END IF

INSERT course into course hashtable

END FOR

END function

### Function printCourseInformation()

START function: void printCourseInformation(courseId)

SET key equal to the hash value of courseId

CREATE new node, printNode

SET printNode equal to the node found at key

IF printNode is not null, its key isn’t the default key and its course ID is equal to courseId

OUTPUT course info for printNode

RETURN

ELSE IF printNode is null or printNodes key is the default key

OUTPUT “No course found with this ID”

RETURN

END IF

WHILE printNode is not null

IF printNode’s key is not the default key and its course ID is equal to courseId

OUTPUT course ID and name for printNode

IF printNode’s course’s prerequisite list is not null

FOR each item in the prerequisite list

OUTPUT current prerquisite’s course’s course ID

END FOR

END IF

RETURN

ELSE

SET printNode equal to its successor

END IF

END WHILE

END function printCourseInformation