1. **Design pseudocode to define how the program opens the file, reads the data from the file, parses each line, and checks for file format errors.**

FUNCTION readFile(File f, lines[])

courseNumbers[], courseTitles[], prerequisites[], line

i = 0, j = 0

Flag = TRUE

WHILE (NOT END OF FILE f)

courseInfo[] = SPLIT (READLINE(f, line), DELIMETER = , )

APPEND line TO lines

IF (LENGTH of courseInfo < 2)

Flag = FALSE

BREAK

END IF

courseNumbers[i] = courseInfo[0]

courseTitles[i] = courseInfo[1]

INCREMENT i

IF (LENGTH of courseInfo > 2)

FOR k = 2 to LENGTH of courseInfo

prerequisites[j] = courseInfo[k]

INCREMENT j

END FOR

END IF

END WHILE

IF Flag == TRUE

FOR each P in prerequisites

IF P NOT IN courseNumbers

Flag = FALSE

BREAK

END IF

END FOR

END IF

RETURN Flag

END FUNCTION

1. **Design pseudocode to show how to create course objects and store them in the appropriate data structure**.

CLASS Course

Number: String

Title: String

Prerequisites[]: String[]

CONSTRUCTOR Course(line)

Number = SPLIT(line, DELIMETER = ,)[0]

Title = SPLIT(line, DELIMETER = ,)[1]

IF LENGTH of SPLIT(line, DELIMETER = ,) > 2

Prerequisites = SPLIT(line)[ 2 to LENGTH of SPLIT (line, DELIMETER = ,)]

END IF

END CONSTRUCTOR

END CLASS

FUNCTION createObject(Courses <Course>, File f)

Lines[] = " "

IF readFile(f, Lines) == TRUE

FOR each Line in Lines

APPEND NEW Course(Line) TO Courses

END FOR

END IF

ELSE PRINT("File cannot be read")

END ELSE

END FUNCTION

1. **Design pseudocode that will print out course information and prerequisites**.

FUNCTION MAIN()

Filename = INPUT()

File F = NEW File(Filename)

Courses <Course> : vector

CALL : createObject(Courses, F)

CourseNumber = INPUT()

IF Courses is EMPTY

PRINT ("No objects read from the file")

END IF

ELSE

printCourseInformation (Courses, CourseNumber)

END ELSE

END FUNCTION