

# C projects 2023

Global variables may not be used in any of the assignments.

Write user friendly programs:

- When user is requested to input a value, the program must clearly state what the user is expected to enter
- Error message shall be displayed if the user enters an invalid value.
- Validate input that is read from a file and display a descriptive error message if data is not valid.
- Comment your code and use meaningful variable and function names to make it more understandable.

Reports must print data in formatted columns where printed data is aligned by columns. For example, when printing numbers make the field width large enough to hold largest number that you are going to print.

Take care that array bounds are not violated - especially with strings!

Do not use “magic numbers” - use `#define` for array sizes and any other limits/constants that you may need.

Start by thinking what type of input you need to ask from the user or read from a file. Write functions that perform error checking for types that you need.

Use functions to structure your program.

## Transcript printer

Write a program that reads course information from a file. The file is a text file where each line contains either course name and number of credits in square brackets or student name and grade separated with a colon. For example:

```
[Computer architecture:5]
Billie Mandala: 5
Arthur Dent: 3
Gabriel José de la Concordia García Márquez: 2
[C programming:3]
Olive Nelson: 0
Dilbert Einstein: 5
Arthur Dent: 4
```

Empty lines are ignored. White space at the beginning or end of the student name or course title is removed. Names that appear before any course title are ignored and error message is printed.

Program asks user to enter students name and then prints a list of courses that the student has taken with grades and finally the average grade. If user enters "stop" as the name the program stops.

## Student grade calculator

Write a program that calculates and displays the final grades for a student based on their scores in different subjects. The program should allow the user to input scores for multiple subjects and then calculate the average grade and overall grade for the student.

Requirements:

- The program should start by asking the user for their name.
- Then, it should ask the user how many subjects they want to calculate grades for.
- For each subject, the program should prompt the user to enter the subject name and the score (as a percentage).
- Calculate the average grade based on the scores and display it.
- Determine and display the overall grade based on the following grading scale:
  - 5: 90-100
  - 4: 80-89
  - 3: 70-79
  - 2: 60-69
  - 1: 50-59
  - 0: Below 50
- The program must also write the same report to a file

```
Welcome to the Student Grade Calculator!
```

```
Please enter your name: John Deere
```

```
How many subjects do you want to calculate grades for? 3
```

```
Enter subject 1 name: C Programming
```

```
Enter your score for C Programming (0-100): 85
```

```
Enter subject 2 name: Python
```

```
Enter your score for Python (0-100): 92
```

```
Enter subject 3 name: Computer Architecture
```

```
Enter your score for Computer Architecture (0-100): 68
```

```
-----  
Student: John Deere  
-----
```

Subject	Score	Grade
C Programming	85%	4
Python	92%	5
Computer Architecture	68%	2

```
Average Grade: 3.67  
-----
```

## CSV reader

CSV (Comma-Separated Values) files are a common way to store tabular data, making them a valuable format for data interchange between applications. Write a program that reads house temperature data from a CSV-file and prints a horizontal bar graph of the temperature in the selected room. The CSV file starts with a header row and then continues with lines of comma separated data. The header row contains two titles: Room and Temperature. Data lines contain room name and the temperature separated by commas.

```
Temperature,Room
22.5,Kitchen
18.7,Living Room
24.2,Bedroom
20.1,Kitchen
12.3,Living Room
23.8,Bedroom
16.9,Kitchen
19.4,Living Room
13.7,Bedroom
21.8,Kitchen
11.5,Living Room
24.9,Bedroom
```

Program must ask user to select a room and then print the temperatures with one decimal precision followed with a bar graph using dashes (-). Each dash corresponds to 0.5 centigrade and temperatures in the range of 0 – 30 are printed as horizontal bars. Temperatures that are outside or the range are printed as a single X on the left. If the selected room does not exist then program must print an error message.

For example, (not based on the data above):

```
Bedroom
20.4 -----
22.5 -----
21.4 -----
21.6 -----
32.3 X
18.2 -----
18.8 -----
19.5 -----
21.3 -----
```

## Study planner

Write program to print your daily schedule. The program reads your schedule from a CSV file. The header row has four headings: Day, Time, Course, Room. Data rows need to appear in any specific order. Program asks user to choose a day and prints the day's schedule starting with earliest time in ascending order. If there are no classes for the day program prints "No classes today". If user enters "stop" as a day the program stops.

```
Day, Time, Course, Room
Monday, 9:00, C programming, KNE555
Tuesday, 13:00, Network, KMD666
Wednesday, 10:00, Academic writing, KMD777
Monday, 13:00, Math, KME663
Friday, 14:00, Routing, KMD666
Tuesday, 9:00, Java, KNE556
Friday, 8:00, Online course, N/A
```

For example:

```
Enter day: Tuesday
On Tuesday you have:
  9:00 Java, KNE556
 13:00 Network, KMD666
Enter day: Thursday
No classes today
```

## Repository manager

Write a program for keeping track repositories you use. The program records aliases (short names) for long repository links and allows user to view and manage the aliases.

The programs ask user what to do and parses the user input for supported commands. The commands are:

- add <alias> <repository link>
  - adds a new entry
- show <alias>|all
  - user can show the repository link of a given alias or show all links
- list
  - prints a list of all aliases
- quit
  - ends the program
- delete <alias>
  - deletes the given entry
- help
  - prints a list of available commands

If the command is not recognized an error message must be printed.

Program must read the information from a file when program starts and must save the changes to the file either immediately after change or when program ends.