Thessaloniki, 14/01/2024

Tel: +30 231 039 8384
Fax: +30 231 039 8389
E-mail: astaras@act.edu

COURSE PROJECT DESCRIPTION

COMPUTER SCIENCE 230

Introductory Systems Programing with C++

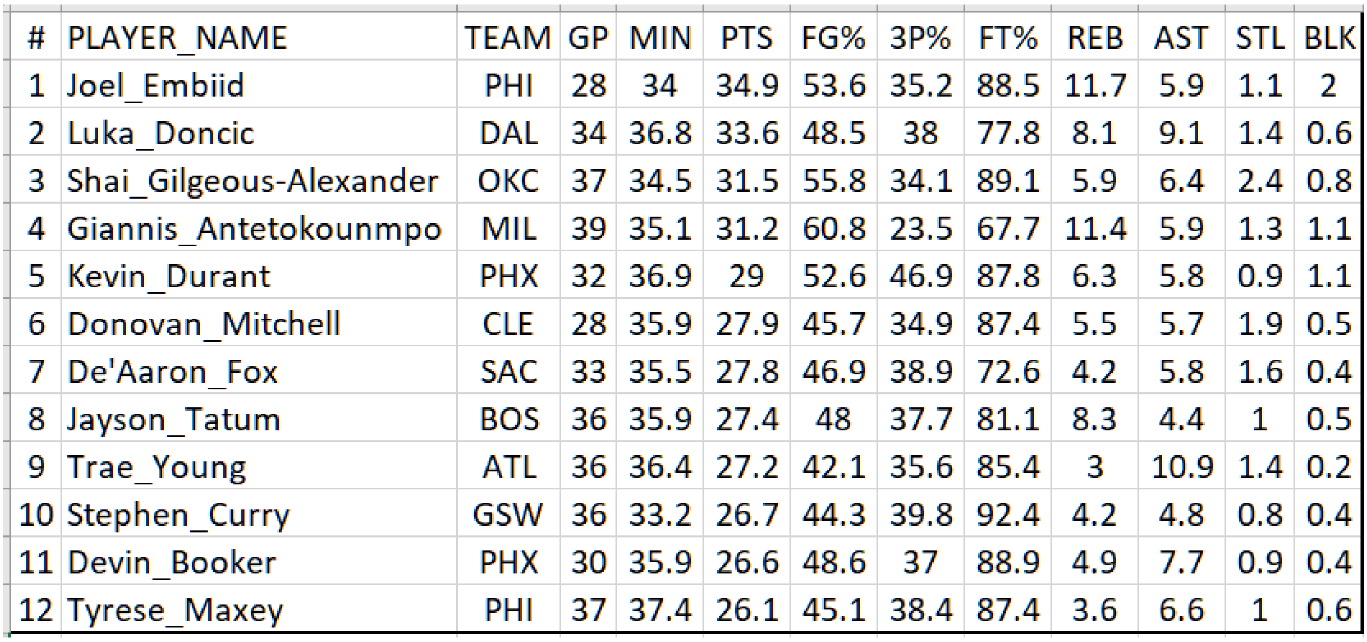
C++ Programing Project

This document contains a description of your CS230 course project, worth 30% of the total grade. It involves coding a C++ program which manipulates statistics data regarding the top performing players in this season in the NBA, the National Basketball Association in the USA.

Upload your **C++ code (80%)** as well as a brief **2-page project report (20%)** to the relevant node on the course Moodle page. The report should contain a concise list of your *design requirements*, a *function interconnectivity diagram*, a top-level (i.e. not very detailed) *flowchart diagram,* a couple of explanatory *screenshots* and a brief *bibliography* section. All items should be part of the report PDF file, except for the C++ code which should be uploaded as a separate .cpp file.

Briefly research the 2023-24 NBA season online. If you have never played or watched basketball being played, try to gain a basic understanding of fundamentals, as this will likely prove valuable when planning for this project. Try spending a few minutes reading through some info web pages.

Download the comma separated volume (CSV) file *NBA_2024_stats.csv* from the course Moodle page. Import the file to your favourite spreadsheet application and view the data. It is sorted by total points scored per game (6th field) as shown in the picture below and contains data about this season's 12 top performing players. The file is in the CSV format, which means it's a text data file: open it using a text editor and examine it. Your C++ project code will need to read this file and manipulate the data it contains.



#	PLAYER_NAME	TEAM	GP	MIN	PTS	FG%	3P%	FT%	REB	AST	STL	BLK
1	Joel_Embiid	PHI	28	34	34.9	53.6	35.2	88.5	11.7	5.9	1.1	2
2	Luka_Doncic	DAL	34	36.8	33.6	48.5	38	77.8	8.1	9.1	1.4	0.6
3	Shai_Gilgeous-Alexander	OKC	37	34.5	31.5	55.8	34.1	89.1	5.9	6.4	2.4	0.8
4	Giannis_Antetokounmpo	MIL	39	35.1	31.2	60.8	23.5	67.7	11.4	5.9	1.3	1.1
5	Kevin_Durant	PHX	32	36.9	29	52.6	46.9	87.8	6.3	5.8	0.9	1.1
6	Donovan_Mitchell	CLE	28	35.9	27.9	45.7	34.9	87.4	5.5	5.7	1.9	0.5
7	De'Aaron_Fox	SAC	33	35.5	27.8	46.9	38.9	72.6	4.2	5.8	1.6	0.4
8	Jayson_Tatum	BOS	36	35.9	27.4	48	37.7	81.1	8.3	4.4	1	0.5
9	Trae_Young	ATL	36	36.4	27.2	42.1	35.6	85.4	3	10.9	1.4	0.2
10	Stephen_Curry	GSW	36	33.2	26.7	44.3	39.8	92.4	4.2	4.8	0.8	0.4
11	Devin_Booker	PHX	30	35.9	26.6	48.6	37	88.9	4.9	7.7	0.9	0.4
12	Tyrese_Maxey	PHI	37	37.4	26.1	45.1	38.4	87.4	3.6	6.6	1	0.6

An additional file named *NBA_2024_stats.xlsx* is supplied: it is a MS Excel spreadsheet file which contains helpful links to the nba.com website to help you better understand the data. If you need Microsoft Excel user support, free tutorials for the 2019 version can be found here.

Build a C++ code which performs the following tasks:

- Presents an explanatory greeting and 7-option menu, capable of trapping erratic user input. Once each task related to an option is completed, the program should redisplay the menu options. (**5%**)
- **Option 1:** Reads the 13 columns of data from file NBA_2024_stats.csv and displays it. (**10%**)
- **Option 2:** Asks the user for a lower limit of average points scored per game, then filters out players below the limit (i.e. displays the name, team name and average points per game only for players who have an average PTS equal to or greater than the limit). (**10%**)
- **Option 3:** Allows the user to enter data for a new player (a value for each data field). The data should be stored into as a new version of file *NBA_2024_stats.txt* (overwrite the old one.) (**10%**)
- **Option 4:** Asks the user for a field to sort by, then whether the order should be ascending or descending. Reads the original 13-column data from file *NBA_stats.txt,* then displays it on screen, sorted as the user requested. It also saves the sorted data into a file named *NBA_sorted_stats.txt .* (**20%**)
- **Option 5:** Reads the 14 columns of data from file *NBA_sorted_stats.txt* and displays it on screen. The program should check whether the file exists, and if it doesn't it should prompt the user to run option 4 first. (**10%**)
- **Option 6:** Calculates a new performance metric (NEW) = (PTS) x (REB) x (AST) x (0.001). The new metric data value should be rounded to 1 decimal point; stored in a new (14th) column in a file called *NBA_processed_stats.txt;* read back into memory; and subsequently all player records should be displayed on the screen sorted by the value of this new metric. (**15%**)
- **Option 7:** Exit the program.

```
 =========================================================================
 USER MENU: TOP SCORER BASKETBALL STATISTICS FROM THE 2023-2024 NBA SEASON
 =========================================================================
1. Read and display NBA stats for the top 12 players
2. Filter data by points scored per game
3. Enter data for an additional player
4. Sort the player records and save them into a new file
5. Display the sorted data
6. Calculate a new performance metric (NEW), display sorted data
7. Exit the program
```

Your program should ensure that every time the user is asked to make a menu choice, they have the required data available to read on screen, with clear column labeling, as well a listing of all menu option descriptions. Every time data is saved into a file, the user should be notified.

If you face difficulties with available screen width, you may skip displaying data which is not related to the relevant menu option.

You may do some additional relevant development for **up to 10% extra credit** in this assignment. Here are a few ideas:

- Research how to scrape data off web pages. Build a new menu option which -when activated- causes your code to automatically import fresh data from nba.com about the season’s current top 12 scorers. Save the data into a file and use it to feed all other menu options, instead of using the CSV file supplied on Moodle. (**10%**)

- Research how to create visual graphs using C++ libraries. Import the relevant library, then add a menu option to display bar charts visualizing the data. (**10%**)
- Build a graphical user interface for the menu. (**5%**) Try to make it clickable. (**5%**)

To get maximum points in this project your report should be carefully and concisely written, while your C++ source code should:

- Be well commented.
- Be clear, well spaced and easy to read.
- Clearly prompt the user for I/O.
- Have all variables declared and initialized.
- Possess intuitive variable identifiers (e.g. myFavColour).
- Avoid misuse or waste of programing resources.
- Predict erratic user input, trap it and handle it appropriately.
- Execute in the Dev-C++ IDE without the need to download and/or update non-standard libraries
- Include any necessary data files in the submitted project folder

You may copy portions of code downloaded directly from the CS230 Moodle page (i.e. course material), but not from any of the websites linked there. You may however use code you find on the linked websites as guidance to compose your own. Furthermore, you may use code you find in books, on the internet or other external resources as guidance and inspiration, but you may not copy it directly. In either case, an acknowledgement should be placed in the header comments section of your code and in the bibliography section of your project report.

You may form study groups, work together and exchange opinions with fellow students, however each student should submit their own code and project report, written entirely by themselves.

Sources: “Season Leaders”, https://stats.nba.com/leaders/, [accessed: 15/01/2024]

This assignment assesses the following learning outcomes as mentioned in the syllabus:

A1: Identify and select the appropriate program design strategies to solve simple informatics problems using C++
B1: Demonstrate understanding of the theory of programming: compilers & algorithms
B3: Decompose complex problems to simpler tasks through stepwise refinement
D2: Theory of Programing: Perform problem Decomposition – Stepwise Refinement