

# Analyzing King Saud University Computer Science Curriculum

Mohand Alrasheed

2022 / January

# Contents

**1 Introduction**

**2 Assumptions**

**3 Criteria**

**4 Method**

4.1 Data . . . . .

4.2 Score calculation . . . . .

**5 Results**

5.1 Visuals . . . . .

5.2 Observations . . . . .

**6 Conclusion**

6.1 Code & Demo . . . . .

# 1 Introduction

King Saud University's computer science curriculum has many pros and cons, in this document we outline a criteria for judging courses and ranking the curriculum based on experts' opinions.

## 2 Assumptions

We make a number of assumptions regarding the evaluation of courses, these guide the evaluation process so they should be read carefully before moving on to results.

### Assumptions

1. The course instructor's influence is minimized when ranking courses by imagining all courses being taught by a perfect instructor.
2. The ranking has been chosen by students mostly on their final semester, which would mean most of them have not studied the last semester and in turn we will not take its courses into consideration in this version.
3. The ranking has been chosen by students mostly belonging batch 439 and 438, different batches might rank courses differently as course content changes slowly.
4. Most electives are not taken into consideration, but a small number of popular electives will be considered.

## 3 Criteria

Courses are judged based on five criteria chosen by the most elite members of 439.

### Criteria

1. **Applications:** This refers to real-world applications of the knowledge gained by studying the course.

2. **Relevance:** This refers to the new-ness of the knowledge taught compared to the current (as of this document’s date) state of the art.
3. **Insight:** This refers to the quality of the knowledge gained with respect to understanding the world and expanding one’s horizons.
4. **Understanding:** This refers to the proportion of the course’s understanding portions over the memorization portions.
5. **Ease:** This refers to how easy the course was.

## 4 Method

### 4.1 Data

The data is collected and sorted using *Google Forms* in which each reviewer identifies their university batch and sex as well as rate an optional number of courses, i.e. the reviewer can review zero courses, or all courses if they wish. Each reviewer can optionally also provide extra notes alongside any course.

The number of courses ended up being 42, where we include the core plan as well as popular electives. We end up with a number of reviewers totaling 16 reviewers where the minimum number of reviews per course is 2 and the maximum is 14, the distribution of counts is skewed to the right where the 50th percentile is 11 and the 75th is 13. After collecting the data we end up with (Courses, Criteria) X Reviewer matrix.

### 4.2 Score calculation

Due to the low sample size on reviews of some courses we treat the problem as a *Bayesian estimation* problem. In our implementation we possess an idea of the original values we’re predicting what we call the *prior*, which in this case will be the arithmetic mean of a criterion on some category<sup>1</sup>, then we update our beliefs given new evidence (reviews) by taking a weighted average

---

<sup>1</sup>Categories are divided in this way: Humanities, Mathematics, Chemistry, Islamics, Physics, Computer science

of the criterion and the prior.

$$(r_1 * w_1 + .. + r_n * w_n + c_{mean} * w_c) / (\sum_{i=1}^n w_i + w_c)$$

We experimented with the weight of the category mean  $c_{mean}$  and arrived at a weight half as weighty as the weight of the reviews.

After collecting and processing the data we will rank courses using a weighted mean of criteria, we will use three different weights.

### Weighing technique

1. **General:** This weighing takes everything into account fairly, equivalent to a traditional mean.

$$Applications + Relevance + Insight + Understanding + Ease$$

2. **Real-world:** This score mainly focuses on real world utility.

$$Applications*1.5 + Relevance + Insight*0 + Understanding*0.2 + Ease*0.5$$

3. **Academic:** This score only focuses on the academic aspect of courses.

$$Applications*0 + Relevance*0 + Insight*1.5 + Understanding*1.5 + Ease*0$$



## 5 Results

	Real-world score	Academic score	General score	categories
ENGLISH100	0.977969	0.587055	0.749899	Humanities
ARB100	0.644273	0.561107	0.559423	Humanities
MATH101	0.899391	0.883593	0.772007	Mathematics
CHEM101	0.520728	0.438133	0.445344	Chemistry
STAT101	0.927789	0.804526	0.756429	Mathematics
TECH101	0.765262	0.545605	0.626730	Humanities
ENTREPRENEUR101	0.601895	0.521720	0.530546	Humanities
FAJAB101	0.600165	0.538924	0.546742	Humanities
NAHAJ101	0.638404	0.661528	0.594242	Humanities
ENGLISH110	0.918583	0.582118	0.713858	Humanities
SALAM107	0.754027	0.638643	0.659744	Islamics
PHYS104	0.624921	0.534890	0.519392	Physics
MATH106	0.794863	0.735696	0.674323	Mathematics
CSC111	0.984843	0.934987	0.849246	Computer science
MATH151	0.954951	0.868024	0.817721	Mathematics
SALAM108	0.634832	0.415596	0.526326	Islamics
CSC113	0.840633	0.775292	0.730219	Computer science
CSC220	0.862635	0.832111	0.739635	Computer science
MATH244	0.831908	0.716022	0.674008	Mathematics
CSC212	0.943643	1.000000	0.839300	Computer science
CSC215	0.801212	0.781024	0.676474	Computer science
MATH281	0.848302	0.846393	0.748259	Mathematics
CSC304	0.676360	0.582436	0.612944	Computer science
CSC380	0.795881	0.727745	0.676944	Computer science
CSC227	0.740514	0.637227	0.619513	Computer science
CSC311	0.993665	0.991418	0.866180	Computer science
CSC339	0.671742	0.846109	0.679513	Computer science
CSC343	0.659292	0.501000	0.529513	Computer science
CSC361	0.801822	0.819164	0.703416	Computer science
CSC329	0.971137	0.964173	0.849513	Computer science
CSC340	0.609492	0.737127	0.586180	Computer science
CSC453	0.658107	0.646309	0.582847	Computer science
CSC496	0.898492	0.796582	0.736309	Computer science
PHYS103	0.611916	0.635857	0.542905	Physics
CSC443	0.683071	0.549409	0.578540	Computer science
CSC462	1.000000	0.922291	0.843796	Computer science
CSC489	0.792190	0.732545	0.691387	Computer science

## 5.1 Visuals

We include the most important visualizations done. Which are the general scores ranking and the weighted scores ranking. More visualization and a live demo with interact-able graphs can be found here: [courses.hawzen.me](https://courses.hawzen.me).



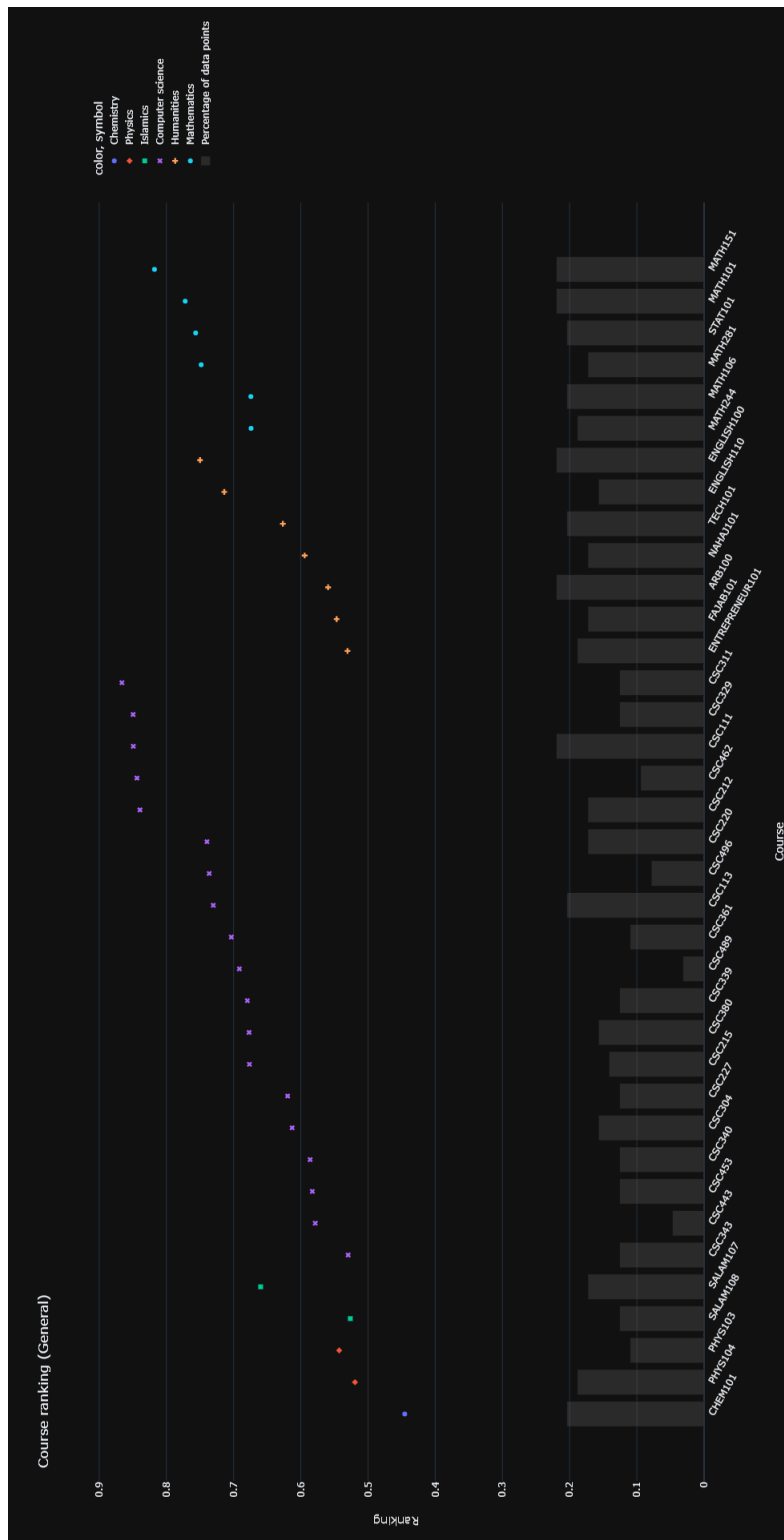


Figure 1: Scores based on general ranking

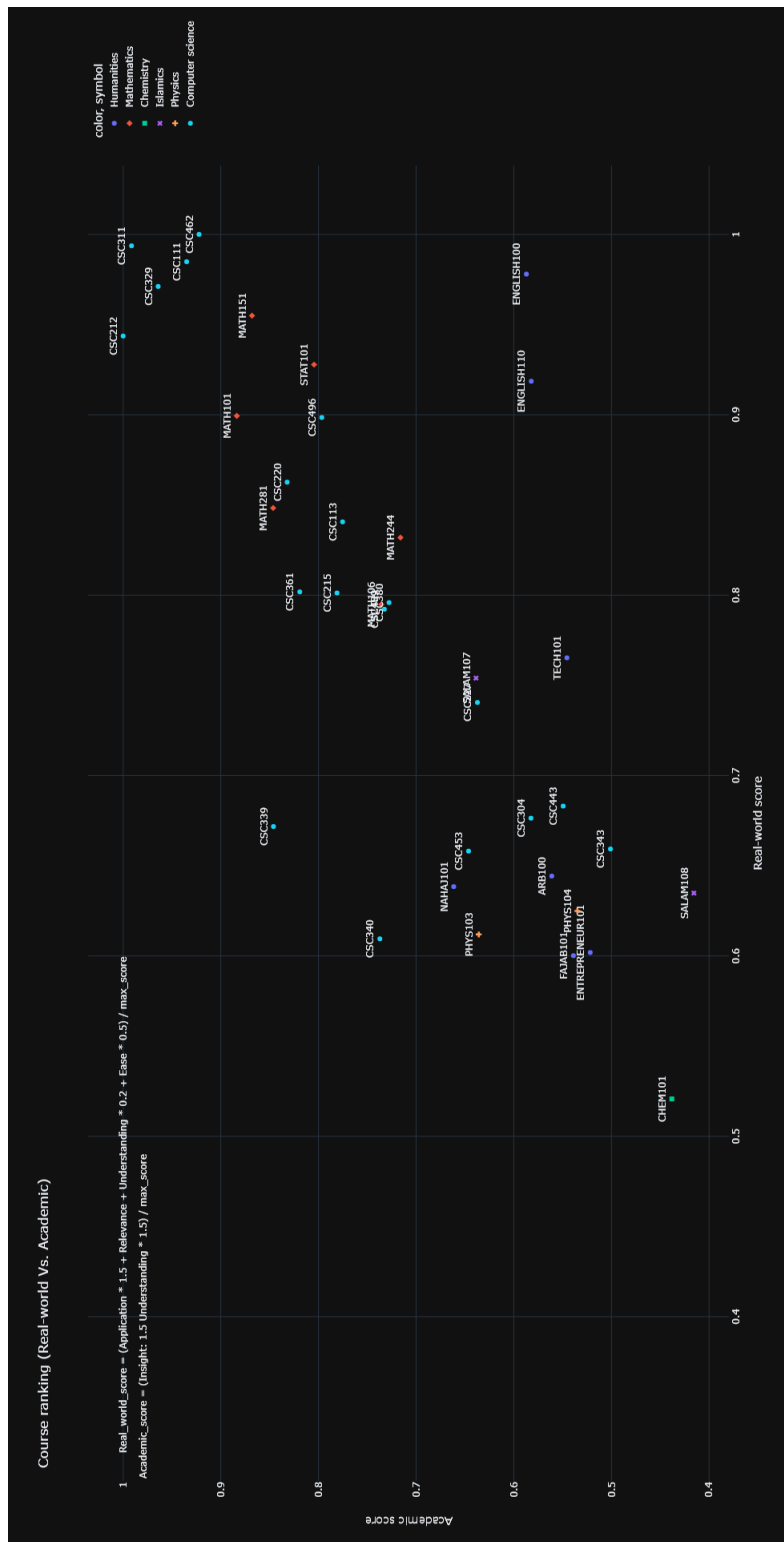


Figure 2: Scores based on Real-world Vs. Academic ranking

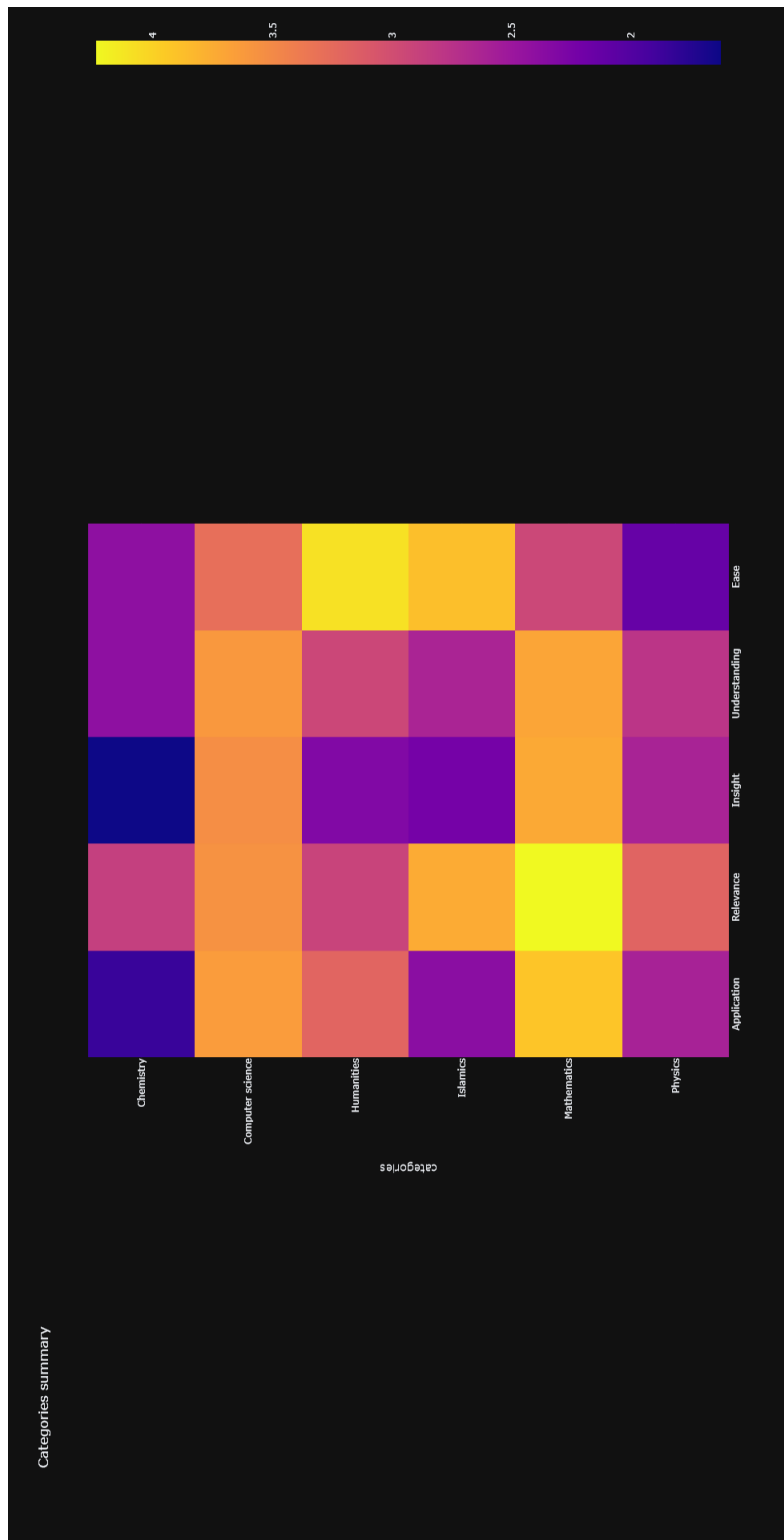


Figure 3: Scores based on Real-world Vs. Academic ranking

## 5.2 Observations

We will detail a couple observations from the visualizations.

### Figure 1

1. The top three courses in general ranking are CSC311, CSC329 and CSC111. All of which are core computer science and are highly applicable.
2. The bottom three courses in general ranking are CHEM101, PHYS104 and SALAM108. All of which are not computer science related, and are all studied in high school already.
3. The lowest ranking Math courses are MATH106 and MATH244, probably due to it's unrelated nature compared to other math courses like MATH281 and MATH151 as well as being harder than all math courses.
4. The lowest ranking computer science course is unsurprisingly CSC343 (Intro to Software Engineering).

### Figure 2

1. We can see a general linear pattern in the scatter plot, which tells us that having a high academic score is intimately correlated with having a high real-world score.
2. Outliers which have a high Academic score but a low Real-world score are CSC339 (Theory of computation) and CSC340 (Compilers and translation).
3. Outliers which have a high Real-world score but a low Academic score are the two English courses, which makes sense, as English is extremely useful in the real world, but isn't academically related to computer science.

### Figure 3

1. Computer science and Mathematics categories are highly correlated in all criteria, though we can see that mathematics have a higher score in relevance but a lower score in Ease. This makes sense if one considers that math is slowly changing, and courses like calculus are up to date, where courses like CSC453 (Parallel processing) and CSC380 (Databases) have developed so much compared to how they're studied in class.
2. Humanities are generally easy, un-Insightful and its content is not up to date, similar to Islamics, with the exception of Relevance.
3. Chemistry has extremely low values for all criteria, probably because the category is very distant from computer science and the category has only one course in it.

## 6 Conclusion

After looking at student opinions on all courses, we compiled a number of suggestions to alter the computer science plan in hopes of raising the scores.

**Low Relevance scores** A few courses had excessively low Relevance scores, this means students think these courses are very old and need updating. Courses under 3 in relevance are:

1. ARB100
2. TECH101<sup>2</sup>
3. Entrepreneure101.
4. FAJAB101
5. NAHAJ

---

<sup>2</sup>This course has been overhauled recently, and most reviewers have not studied the overhauled course

6. CHEM101
7. CSC380
8. CSC343
9. CSC453
10. CSC443

Low relevance score in core computer science courses like CSC380 and CSC453 is not good, and an easy way to improve those courses is by updating the material. CSC380 for instance is only concerned with relational databases, where in the real world so many database variants are out there but the course doesn't even mention them as it created ages ago.

**High Application scores** The highest computer science Application scores are doing something right, we will mention their names and detail what those courses do differently from other courses.

1. CSC111 (Programming I)
2. CSC212 (Data Structures)
3. CSC311 (Algorithm analysis and design)
4. CSC462 (Machine learning)

All courses mentioned had many Labs/Homeworks/Programming Assignments, other courses with very low application scores had relatively few or no programming assignments, thus, we can see that this factor is highly correlated with the application score. This doesn't mean imposing assignments on courses like CSC304 (Ethics) is going to be helpful though.

**Low Understanding scores** Many courses have low Understanding scores (i.e. rely heavily on memorization) such as:

1. CSC343 (Intro to Software Engineering)
2. CSC227 (OS)

### 3. CSC443 (IT Project Management)

This is not entirely due to the course's nature, as courses like Networking which have a large theoretical body to read scored very highly on this criterion. Carefully chosen questions on exams and quizzes can change the perception from memorization to understanding. This is generally unfavorable in all circumstances and is possibly an area where change could improve the computer science plan.

## 6.1 Code & Demo

The code can be found at [github.com/Hawzen/Course-Ranking](https://github.com/Hawzen/Course-Ranking)

A live demo can be found at [courses.hawzen.me](https://courses.hawzen.me).