UIL COMPUTER SCIENCE THE FIRST 15

Introduction

This guide is written for students preparing to take the Texas University Scholastic League Computer Science written exam. The first fifteen questions of each UIL Computer Science written exam are purposely more basic (easier) than the remaining twenty five questions. This guide is intended to show you how to correctly answer the first fifteen questions. UIL provides a list of topics to be covered by the written exam at the following web location: http://www.uiltexas.org/files/academics/UILCS-JavaTopicList1516.pdf. Within that document can be found a list of topics used to create the first fifteen questions on the test. Those fifteen topics will serve as the chapters for this guide.

A goal of this guide is to help students that are just beginning their computer science studies prepare to be competitive at a basic level. Each correct answer on the Computer Science written exam is worth six points. Each incorrect answer deducts two points from the score. Skipped questions are not penalized. So, if you can answer the first fifteen questions correctly, you will earn a score of ninety (90) points. ***In 2015 a score of ninety would have earned a medal (6th place or better) at the district level 86 percent of the time across all divisions.*** Not only does 6th place earn a medal for the individual competitor, 6th place earns four points towards the team academic championship for your school. Here is a chart that shows the percent of district contests where a score of 90 would have earned a district medal.

Percent of district Computer Science contests where a score of 90 or above would have earned a medal (6th place or better).	
Classification	**Percent**
1A	100%
2A	100%
3A	100%
4A	97%
5A	75%
6A	47%

UIL Computer Science is not just an individual competition. There is a team component as well. Four students compete on a team and one half of the team score is comprised of the total of the top three scores on the written exam. The remaining half of the team score comes from the team members' performance in the hands-on (programming) portion of the contest. How to succeed at the programming portion of the contest will be left for another guide.

To further illustrate the importance of successfully answering the first fifteen questions consider that if three members of a team can each score 90 points by correctly answering the first 15, that team's written score will be 270 points. That is 270 points WITHOUT any points from the programming portion of the contest. To illustrate the importance of 270 points on the written exam here is a chart that shows the percentage of districts where a team score of 270 or more would have placed a team 3rd or better at a district contest including programming.

Percent of district Computer Science contests where a team score of 270 or above would have earned a team medal (3rd place or better).	
Classification	**Percent**
1A	100%
2A	100%
3A	100%
4A	88%
5A	72%
6A	34%

The reason 3rd place has been chosen to be included in the chart is that Computer Science and CX Debate are the only two UIL Academic contest for which team points and team medals are given for 3rd place. Points are given as follows: 1st place team 20 points, 2nd place team 16 points, 3rd place team 12 points. It is easy to see that simply mastering the first fifteen basic concepts on the test can produce great results for you, your Computer Science team, and your school.

A few words about what this guide is NOT are appropriate at this time. This is a GUIDE, not a textbook. This guide is not a complete and exhaustive discussion of the topics presented. Reading and understanding this guide is NOT intended as a course in computer science. Using this guide along with enrollment in a good Computer Science 1 course will enhance your success without a doubt. The more that you seek out and use alternative resources that are available (many are listed in Appendix B), the more successful you will become. This guide will not address the topics or complexity needed to answer the remaining 25 questions on each written exam. To go beyond district and be successful at region and state will most definitely require that you gain a much deeper understanding of ALL of the topics covered by the entire test.

Success in the UIL Computer Science competition requires a great deal of hard work and dedication, even to just get the first fifteen questions correct! To be successful you will have to practice A LOT! To be successful you and your teammates will have to memorize A LOT! But, if you will read and study this guide, memorize what you are asked to memorize, practice often, and make use of all of the resources available to you, you can learn a lot of computer science and maybe win a few medals along the way.

Let’s get started!

Number Base Concepts

For many years the first question on every test has been about changing bases. Changing from decimal to binary, binary to hexadecimal and so on. Usually the question involves doing some arithmetic using values presented in different bases and the answer choices will be in other bases as well. Recently the questions have asked which of the answer choices are NOT equivalent to the expression shown. This requires you to convert all of the answer choices to the same base as the answer you have calculated.

Each place in a decimal number represents a power of 10. For example 3628 is 3 one thousands, 6 one hundreds, 2 tens and 8 ones. Or looked at another way:

$8 \text{ X } 10^0 = 8$

$2 \text{ X } 10^1 = 20$

$6 \text{ X } 10^2 = 600$

$3 \text{ X } 10^3 = 3000$

3000 + 600 + 20 + 8 = 3628

While it is certainly the most common way, using 10 as a base is not the only way to represent a value. In fact any number can be used as a base. In computer science the three most common alternative numbers to use as a base are 2, 8 and 16. The UIL CS test will also sometimes test your ability to covert numbers to other bases as well.

When a number is represented in a base other than 10, the base will be shown as a subscript at the end of the number. For example, 101010_2 is base 2 and 123_8 is base 8.

To be able to correctly answer the number base problems you are going to have to be able to convert a number like 101010_2 to its decimal equivalent of 42 and the other way around.

Converting From Binary to Decimal

In the binary number system there are only 2 digits. They are 0 and 1. Each digit in a binary number represents a power of 2 starting with 2^0 as the rightmost digit and increasing as you go to the left. Here are the first eight place values:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

To convert from a binary number to a decimal number, for each binary place that has a 1 as its digit add the decimal place value.

Here are a few examples:

$101_2 = 5$

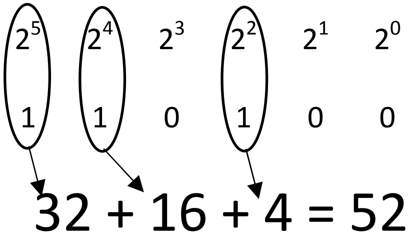
There is one 1, no 2's and one 4. 1+4=5.

110100 = 52

There are no 1's or 2's, one 4, no 8's, one 16 and one 32.

4 + 16 + 32 = 52

Here is an illustration that might be helpful.



Here is a more structured way to look at it. Consider this binary number.

10110110

Now draw two rows of boxes and write the place value for each binary digit across the top row. Then place the binary number in the second row

128	64	32	16	8	4	2	1
1	0	1	1	0	1	1	0

If there is a 1 in a box in the second row, add the corresponding place value in the top row.

128 + 32 + 16 + 4 + 2 = 182

So, $10000110_2 = 182_{10}$

Converting From Decimal to Binary

What about the other way around. How can we convert the decimal number 81 to a binary number?

One way is to reverse the process from above using the two rows of boxes. Once again draw two rows of boxes. The number of boxes you draw really only needs to go up to enough to hold the largest power of 2 that is still smaller than the decimal number you are trying to convert. In this case that power of 2 is 64.

64	32	16	8	4	2	1