



IB Computer Science SL

Pre-requisites: Computer Science Pre AP/IB



Instructor: Mr. Barry Wagner

E-Mail Address: bwagner@dentonisd.org

Conference Period: A2, B4

Phone: 940-369-2067

Tutorials: M-F 8:10 – 8:50

Course Description

Computer Science SL emphasizes the object-oriented programming methodology introduced in the Pre-IB course with a concentration on problem solving, algorithm development, and program design.

Resources

Web-based curriculum (<http://bwagner.org>)

Aims and Objectives

It is in this context that the Diploma Programme computer science course should aim to:

1. provide opportunities for study and creativity within a global context that will stimulate and challenge students developing the skills necessary for independent and lifelong learning
2. provide a body of knowledge, methods and techniques that characterize computer science
3. enable students to apply and use a body of knowledge, methods and techniques that characterize computer science
4. demonstrate initiative in applying thinking skills critically to identify and resolve complex problems
5. engender an awareness of the need for, and the value of, effective collaboration and communication in resolving complex problems
6. develop logical and critical thinking as well as experimental, investigative and problem-solving skills
7. develop and apply the students' information and communication technology skills in the study of computer science to communicate information confidently and effectively
8. raise awareness of the moral, ethical, social, economic and environmental implications of using science and technology
9. develop an appreciation of the possibilities and limitations associated with continued developments in IT systems and computer science
10. encourage an understanding of the relationships between scientific disciplines and the overarching nature of the scientific method.

Students will be expected to fulfill the following objectives:

1. Know and understand:
 - a. relevant facts and concepts
 - b. appropriate methods and techniques
 - c. computer science terminology
 - d. methods of presenting information.

2. Apply and use:
 - a. relevant facts and concepts
 - b. relevant design methods and techniques
 - c. terminology to communicate effectively
 - d. appropriate communication methods to present information.

3. Construct, analyse, evaluate and formulate:
 - a. success criteria, solution specifications including task outlines, designs and test plans
 - b. appropriate techniques within a specified solution.

4. Demonstrate the personal skills of cooperation and perseverance as well as appropriate technical skills for effective problem-solving in developing a specified product.

IB Assessment

Student assessment is carried out by a combination of an external examination conducted at the end of the course, and an internal assessment, carried out by the teacher. The following tables show the approximate percentage weighting in a typical examination session for each of the assessment objectives across each of the components.

Assessment objective	Paper 1	Paper 2	Internal assessment	Overall
1. Demonstrating knowledge and understanding	24	13	9	46
2. Applying and using	13	7	8	28
3. Constructing, analysing, evaluating and formulating	8	5	4	17
4. Using skills	n/a	n/a	9	9
Component weighting	45%	25%	30%	100%

Internal Assessment (IA)

As mentioned above a large part of a student's assessment is a component called the Internal Assessment (IA). In this course the IA is a computer program that a student must design and implement for a specific client. The program must be developed using the following criterion.

Criterion
Criterion A: Planning
Criterion B: Solution overview
Criterion C: Development
Criterion D: Functionality and extensibility of product
Criterion E: Evaluation

There will be specific due dates for each of the 5 criterion. If a student does not complete the criterion by the assigned due date mandatory tutorials will be assigned until the student completes the work. A significant amount of class time will allotted for students to work on their IAs.

Grading Policy

Major Summative Assessment	70%
- Unit Tests, Programming Projects	
Minor Summative Assessment	30%
- Lab Exercises, Quizzes, IA rough drafts	
Formative Assessment	0%
- Worksheets	

Rules/Class Procedures

1. Be on time.
2. Complete labs in the time allotted.
 - If a student falls more than 1 unit behind the pace of the course mandatory tutorials will be assigned until the work is completed.
3. This Computer lab is for learning computer science and computer programming.
 - Students must complete all assignments before doing work for other classes.
 - Students must complete all assignments before playing on the Internet.
 - Students may only use cell phones the last **10 minutes** of class or if they have completed all assignments.
 - An assignment is not complete until it is recorded on teacher's clipboard or placed in the assignment box.

4. Do not install any software on a computer in this lab.
5. Do not use a **proxy server** to by-pass school district filtering.
6. Stay in your seat until the bell rings or it is time to dismiss.
7. Peer tutoring is encouraged

Course Content and TimeLine

Unit 1 Number Systems - Number Systems - Number Conversion - Data Representation	Week 1
Unit 2 System Fundamentals - What is a System - System Planning - System Design	Week 2-3
Unit 3 Pseudo code and Flowcharts - Pseudo code - Flowcharts	Week 4
Unit 4 Tracing an Algorithm - Trace Tables	Week 5
Unit 5 Computational Thinking - Thinking Procedurally - Thinking Logically - Thinking Ahead - Thinking Concurrently - Thinking Abstractly	Week 6
Unit 6 Methods Review - Methods - Static Methods - Why Use Methods? - Overloaded Methods	Week 7-9
End of 1st 9 Weeks	
Unit 7 Classes Review - Class Decomposition - Anatomy of a Class - Constructors - Accessor Methods - Mutator Methods - toString Method	Week 10-12
Unit 8 Arrays Review - Arrays - Traversing Arrays - Logical Size - Object Arrays - Object Parameters - Array Parameters	Week 12-14
Unit 9 ArrayList - ArrayList	Week 15-18

<ul style="list-style-type: none"> - Wrapper Classes - Enhanced for Loop - remove Method 	
End of 2nd 9 Weeks	
Unit 10 Inheritance <ul style="list-style-type: none"> - Inheritance - Polymorphism 	Week 19-22
Unit 11 Abstract Classes	Week 23-24
Unit 12 Interfaces <ul style="list-style-type: none"> - Interfaces - Comparable Interface - Polymorphism 	Week 25
Unit 13 Searching <ul style="list-style-type: none"> - Linear Search - Binary Search - Comparable Interface 	Week 26-27
End of 3rd 9 Weeks	
Unit 14 Sorting <ul style="list-style-type: none"> - Sorting - Bubble Sort - Selection Sort 	Week 27-28
Unit 15 Computer Organization	Week 29-30
Unit 16 Networks	Week 31-32
Unit 17 Test Prep	Week 33-34
End of 4th 9 Weeks	