IB Computer Science HL 1

Pre-requisites: Honors IBDP Computer Science

Instructor: Mr. Barry Wagner
E-Mail Address: bwagner@dentonisd.org
Planning Periods: B3, B4
Phone: 940-369-2067
Tutorials: M-Th 4:10 – 4:50

Course Description

Computer Science HL 1 is the first year course for students planning on taking the IB Computer Science HL exam. This course will emphasize the object-oriented programming methodology introduced in the Honors IBDP 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 undcerstanding 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.

Grading Policy

- Refer to the District Approved Grading Policy

Rules/Class Procedures

1. Be on time.
2. Complete labs in the time allotted.
    - If a student falls behind the pace of the course, then after school tutorials are available to help him or her get caught up.
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.c
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 - toStrting Method	Week 10-12
Unit 8 Arrays Review - Arrays - Traversing Arrays - Logical Size - Object Arrays - Object Parameters	Week 12-14

- Array Parameters	
Unit 9 ArrayList - ArrayList - Wrapper Classes - Enhanced for Loop - remove Method	Week 15-18
End of 2nd 9 Weeks	
Unit 10 Inheritance - Inheritance - Polymorphism	Week 19-22
Unit 11 Abstract Classes	Week 23-24
Unit 12 Interfaces - Interfaces - Comparable Interface - Polymorphism	Week 25
Unit 13 Searching - Linear Search - Binary Search - Comparable Interface	Week 26-27
End of 3rd 9 Weeks	
Unit 14 Sorting - Sorting - Bubble Sort - Selection Sort	Week 27-28
Unit 15 Computer Organization	Week 29-30
Unit 16 Networks	Week 31-32
End of 4th 9 Weeks	