IB Computer Science HL 2

Pre-requisites: IB Computer Science HL 1

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

Course Description

IB Computer Science HL 2 emphasizes problem solving, algorithm development, and program design using advanced data structures including stacks, queues, linked lists, binary trees, and hash tables.

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	Paper 3	Internal assessment	Overall
1. Demonstrating knowledge and understanding	21	10	9	6	46
2. Applying and using	12	6	7	5	30
3. Constructing, analysing, evaluating and formulating	7	4	4	3	18
4. Using skills	n/a	n/a	n/a	6	6
Component weighting	**40%**	**20%**	**20%**	**20%**	**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 alotted for students to work on their IAs.

Grading Policy

- Refer to the District Approved Grading Policy

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 6 Two Dimensional Arrays - 2D Arrays - Manipulating 2D Arrays	Week 7-8
Unit 7 Recursion - Recursive Thinking	Week 9

- Recursive Programming	
End of 1st 9 Weeks	
Unit 8 Abstract Data Types(ADT) - ADT - List - Stack	Week 11-13
Unit 8 Abstract Data Types(ADT) - Queue - Tree	Week 14-15
Unit 9 LinkedList - LinkedList - Generics	Week 16-18
End of 2nd 9 Weeks	
Unit 10 Stack	Week 19-21
Unit 11 Queue	Week 21-23
Unit 12 Binary Tree - Binary Tree - Binary Search Tree	Week 24-27
End of 3rd 9 Weeks	
Unit 13 Hash Table - Hashing - Collisions	Week 30-31
Unit 14 Case Study	Week 32-33
Unit 15 Review	Week 34-35