Course Title:	International Baccalaureate (IB) Computer Science Higher Level (HL) 2
Grade Level(s):	12
Length of Course:	Two semesters or equivalent term
Credit:	10 units
Prerequisite:	Passing grade in IB Computer Science HL 1

Course Overview:

International Baccalaureate (IB) Computer Science Higher Level (HL) 2 is the second year of a two-year course focusing on enabling and empowering student innovation, exploration, and the acquisition of further knowledge of computer science. Topics covered include system fundamentals, computer organization, networks, computational thinking, problem-solving and programming, as well as the HL extension topics of abstract data structures, resource management and control. Students will also learn through case studies and study one of the following options: databases, modeling and simulation, web science or object-oriented programming (OOP). Students will learn how to think procedurally, logically, concurrently, abstractly, and recursively, to utilize an experimental and inquiry-based approach to problem-solving and to develop algorithms and express them clearly.

Students may be eligible to earn college credit by taking the IB Computer Science HL Exam at the end of the course.

Schools Offering:	Granada High School
Meets University of California Entrance Requirements:	*Seeking UC/CSU Honors "d" Sciences Requirement*
Board Approval:	*Pending Board Approval*
Course Materials:	N/A
Supplemental Materials:	IB Library Resources Various websites and online audio and video resources Teacher Notes and Handouts IB Teacher Support Material such as sample student work available on MyIB resources

IB COMPUTER SCIENCE HL 2

COURSE CONTENT

Subject aims

Diploma Program computer science students should become aware of how computer scientists work and communicate with each other and with other stakeholders in the successful development and implementation of IT solutions. While the methodology used to solve problems in computer science may take a wide variety of forms, the group 4 computer science course emphasizes the need for both a theoretical and practical approach.

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.

Distinction between IB Computer Science HL (Higher Level) and SL (Standard Level)

While the skills and activities of computer science are common to students at both SL and HL, students at HL are required to study additional topics in the core, a case study and also extension material of a more demanding nature in the option chosen. The distinction between SL and HL is therefore one of both breadth and depth.

Additionally, the HL course has 240 hours devoted to teaching, compared with 150 hours for the SL course.

Unit Content

The following IB-required HL Extension Topic Units and the IB Case Study Unit will be the focus of the course content in *IB Computer Science HL 2*. Teachers may also refer back to the *HL 1* Topic Units of the course as they build upon and refer to prior knowledge. The units will not necessarily be taught in order and may often overlap in the study of a given area. The Unit 5 Option Unit will be taught throughout both years but emphasized more heavily in *HL 2*.

Unit 1: HL Extension: Abstract data structures

- Thinking recursively
- Abstract data structure characteristics
- Linked lists
- Trees
- Applications

Summary of Key Assignments and/or Activities
Students will participate in class discussions, give written responses and create programming examples regarding topics such as:

- Identify a situation that requires the use of recursive thinking.
- Trace a recursive algorithm to express a solution to a problem.
- Construct algorithms using the access methods of a stack or queue.
- Sketch linked lists (single, double and circular).
- Describe how trees operate logically (both binary and non-binary).
- Compare the use of static and dynamic data structures.

Unit 2: HL Extension: Resource management

- System resources
- Role of the operating system

Summary of Key Assignments and/or Activities
Students will complete a project which includes:

- Explain the role of the operating system in terms of managing memory, peripherals and hardware interfaces.
- Outline OS resource management techniques: scheduling, policies, multitasking, virtual memory, paging, interrupt, polling.
- Discuss the advantages of producing a dedicated operating system for a device.

Unit 3: HL Extension: Control

- Centralized control systems
- Distributed systems
- Wireless networking

Summary of Key Assignments and/or Activities
In small groups, students will create and give presentations on a variety of topics such as:

- Discuss and analyze a range of control systems.
- Evaluate different input devices for the collection of data in specified situations.
- Discuss the social impacts and ethical considerations associated with the use of embedded systems.
- Compare a centrally controlled system with a distributed system.

Unit 4: The IB Case study

The case study is a valuable teaching tool that can be used to integrate all aspects of the syllabus. The computer science case study provides the stimulus material for the investigation of a scenario involving current developments and/or issues in computer science. The information obtained will prepare students and form the basis of the requirements for answering the questions in HL paper 3.

The case study is an opportunity to keep the course abreast of current technology by introducing new technical concepts or issues requiring a more in-depth investigation than that required in the rest of the course.

This case study will be provided 12 months before the May examination session so that students can carry out detailed research prior to the HL paper 3 examination at the end of the HL 2 course.

Summary of Key Assignments and/or Activities
Through their investigation of the case study, students will:

- demonstrate an understanding of the computer science concepts fundamental to the system(s) in the case study
- demonstrate an understanding of how the system(s) in the case study work
- apply material from the course syllabus in the context of the case study
- explain how scenarios specified in the case study may be related to other similar local and global scenarios
- discuss the social impacts and ethical issues relevant to the case study
- explain technical issues relating to the case study
- evaluate information that may be gathered from local and global sources including field trips, interviews, primary and secondary research, invited guest speakers and online interviews
- evaluate, formulate or justify strategic solutions based on the synthesis of information from the case study itself, additional research and new stimulus material provided in the examination paper

Unit 5: One of four options to be chosen by the teacher (option continued from HL 1)

- Option A: Databases
- Option B: Modeling and simulation
- Option C: Web science
- Option D: Object-oriented programming (OOP)

Summary of Key Assignments and/or Activities
Option A: Databases and further database models and analysis

- Define the following database terms: table, record, field, primary key, secondary key, foreign key, candidate key, composite primary key, join.
- Outline the use of databases in areas such as stock control, police records, health records, employee data.
- Outline the use of databases in areas such as stock control, police records, health records, employee data.
- Evaluate the use of object-oriented databases as opposed to relational databases.
- Compare the different forms of discovering patterns using data mining.

Option B: Modeling and simulation in communications

- Design test-cases to evaluate a model.
- Discuss the correctness of a model by comparing generated results with data that were observed in the original problem.
- Outline the software and hardware required for a simulation.
- Discuss advantages and disadvantages of using a simulation for making predictions.
- Describe the role of chatbots to simulate conversation.
- Outline the evolution of modern machine translators.

Option C: Web science and analyzing the web

- Identify the characteristics of internet protocol (IP), transmission control protocol (TCP) and file transfer protocol (FTP).
- Outline the principles of searching algorithms used by search engines.
- Discuss the effects of the use of cloud computing for specified organizations.
- Explain the role of graph theory in determining the connectivity of the web.
- Outline the difference between the web graph and sub-graphs.

Option D: Object-oriented programming (OOP)

- Construct unified modelling language (UML) diagrams to represent object designs.
- Construct related objects for a given problem.
- Construct code examples related to static arrays.
- Construct list algorithms using object references.
- Explain the importance of style and naming conventions in code.

California Content Standards – Nine through Twelve – Computer Science

9-12.CS.1 Describe ways in which abstractions hide the underlying implementation details of computing systems to simplify user experiences.
9-12.CS.2 Compare levels of abstraction and interactions between application software, system software, and hardware.
9-12.CS.3 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. (P6.2)
9-12.NI.4 Describe issues that impact network functionality.
9-12.NI.5 Describe the design characteristics of the Internet.
9-12.NI.6 Compare and contrast security measures to address various security threats.
9-12.NI.7 Compare and contrast cryptographic techniques to model the secure transmission of information.
9-12.DA.8 Translate between different representations of data abstractions of real-world phenomena, such as characters, numbers, and images.
9-12.DA.9 Describe tradeoffs associated with how data elements are organized and stored.
9-12.DA.10 Create data visualizations to help others better understand real-world phenomena.
9-12.DA.11 Refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.
9-12.AP.12 Design algorithms to solve computational problems using a combination of original and existing algorithms.
9-12.AP.13 Create more generalized computational solutions using collections instead of repeatedly using simple variables.
9-12.AP.15 Iteratively design and develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.
9-12.AP.14 Justify the selection of specific control structures by identifying tradeoffs associated with implementation, readability, and performance.
9-12.AP.16 Decompose problems into smaller subproblems through systematic analysis, using constructs such as procedures, modules, and/or classes.
9-12.AP.17 Create computational artifacts using modular design.
9-12.AP.18 Systematically design programs for broad audiences by incorporating feedback from users.
9-12.AP.19 Explain the limitations of licenses that restrict use of computational artifacts when using resources such as libraries.
9-12.AP.20 Iteratively evaluate and refine a computational artifact to enhance its performance,

reliability, usability, and accessibility.
9-12.AP.21 Design and develop computational artifacts working in team roles using collaborative tools.
9-12.AP.22 Document decisions made during the design process using text, graphics, presentations, and/or demonstrations in the development of complex programs.
9-12.IC.23 Evaluate the ways computing impacts personal, ethical, social, economic, and cultural practices.
9-12.IC.24 Identify impacts of bias and equity deficit on design and implementation of computational artifacts and apply appropriate processes for evaluating issues of bias.
9-12.IC.25 Demonstrate ways a given algorithm applies to problems across disciplines.
9-12.IC.26 Study, discuss, and think critically about the potential impacts and implications of emerging technologies on larger social, economic, and political structures, with evidence from credible sources.
9-12.IC.27 Use collaboration tools and methods to increase connectivity with people of different cultures and careers.
9-12.IC.28 Explain the beneficial and harmful effects that intellectual property laws can have on innovation.
9-12.IC.29 Explain the privacy concerns related to the collection and generation of data through automated processes.
9-12.IC.30 Evaluate the social and economic implications of privacy in the context of safety, law, or ethics.

Instructional Methods and/or Strategies

Students will have individual computers in class, and will work both independently and collaboratively. The teacher will use a mix of techniques such as direct instruction, examples, demonstrations, class discussions and work time for students as instructional methods and strategies. Students will make choices for various project topics with guidance from the teacher. Following a "spiral curriculum," students will revisit topics at various stages of the class with steadily increasing breadth and depth. All students will be expected to mentor, assist and work collaboratively with each other on some projects. Students' projects and efforts will be respected, shared and encouraged by all members of the class.

Class time will include instruction, as well as students doing independent work and/or research using computers, working individually, working with other students and working one on one with the instructor.

Assessment Methods and/or Tools

Throughout the second year of *IB Computer Science HL*, students will be preparing for the IB exams in Year 2. Students will also begin work on their IB Internal Assessment project of developing a computational solution to a real-world issue or problem of their choice. This project will be completed for submission to IB in Year 2 of the course and will be 30% of the overall IB Exam score.

Students will be assessed in a variety of ways as they work on preparing for the IB Exams and assessments throughout Year 2 of the course:

- *Summative and Formative Assessments*- Students will demonstrate their individual knowledge of the *IB Computer Science HL* course material through practice IB assessments, tests and quizzes on subject matter, and work on various projects throughout the course.
- *Group Assessments*- Students will demonstrate their knowledge of the content as well as communicate and work together in small groups or with a partner on various projects. This will also challenge students' critical thinking skills. These projects will be subject to formative and summative evaluation using the IB criteria.
- *Homework*- Students will practice the content material individually through assigned work and use this work as a basis for reflecting on their learning throughout each unit.
- *Class Questions and Discussions*- As formative assessment, students will respond verbally or with written work to questions posed in class that will ask students to explain their arguments and reasoning behind their work.

Assessment Objectives: Students will be expected to demonstrate and achieve the following assessment objectives by the end of the course:

By the end of the course students will:

1. Know and understand:
   - relevant facts and concepts
   - appropriate methods and techniques
   - computer science terminology
   - methods of presenting information
2. Apply and use:
   - relevant facts and concepts
   - relevant design methods and techniques
   - terminology to communicate effectively
   - appropriate communication methods to present information
3. Construct, analyze, evaluate and formulate:
   - success criteria, solution specifications including task outlines, designs and test plans
   - 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 programming product
5. Show advancement in preparation for the IB Computer Science Exam
6. Complete their Internal Assessment project for submission to IB

Honors Courses

In this course, students will be expected to not only demonstrate high level content knowledge through research, computational design, programming, exams and performance tasks, but will also be required to individually design and carry out their IB Internal Assessment Computational Solution projects. Students will also have an opportunity to express their creativity in choosing the topic and format for the completion and presentation of their IB Internal assessment projects, in creating their own programming solutions.

By studying in the IB, students develop core skills for success at university, including:

- interest and experience in authentic research
- applied critical thinking, inquiry, and problem solving skills
- strong academic language and writing skills
- a sense of international mindedness and cultural understanding

- time management and organizational skills necessary to prepare for the IB exams and complete the projects and performance tasks associated with the major assessments central to the course