

## **Basis International School Shenzhen**

**Instructor:** Neill Mark Lee

**Email:** neill.lee@basisinternationalsz.com

**Student Hours:** Mon. & Tues. 4:30 – 5:30, 6006

**Parent Hours:** Fri. 4:30 – 5:30

---

## **COURSE DESCRIPTION**

The prerequisite for this course is either 90%+ in my Foundation CS A course or a successful completion of a pre-assessment which shows solid understanding of Data Types & Variables, if structures, Short-Circuit Evaluation loops (for & while), strings and arrays (1D & 2D).

“AP Computer Science A is an introductory college-level computer science course. Students cultivate their understanding of coding through analyzing, writing, and testing code as they explore concepts like modularity, variables, and control structures.” (College

---

## **GRADING POLICY**

|                                                                                                              |            |
|--------------------------------------------------------------------------------------------------------------|------------|
| <b>General</b><br>(CJ completion/Following School Rules/Collaboration/Participation/Proof Reading Resources) | <b>10%</b> |
| <b>Minor Classwork</b> (Quizzes, Notes, Programs)                                                            | <b>40%</b> |
| <b>Major Tests</b> (at the end of each unit)                                                                 | <b>50%</b> |

## **GRADING SCALE**

All grades will correspond to the following scale:

A =  
A- =  
B+ =  
B =  
B- =

C+ =  
C =  
C- =  
D+ =  
D =

D- =  
F =

- Each program submitted will be graded as follows:
  - 1 mark per program requirement/specification/FRQ mark point.
  - 1 mark for elegant and efficient style.
  - 1 mark for formatting:
    - Meaningful and correctly formatted names.
      - Variable names should begin with lower case letter and in case of multiple words each word after the 1st must begin with a capital letter.
        - e.g. costProd1, totalCost
      - Program names formatted as follows:
        - Unit Number Meaningful Name Your Full English Name.
      - e.g. 02 Add Three Numbers Joe Bloggs
    - Code indented correctly.
  - 1 mark for any comments if required.
- Each set of answers to questions/notes will be marked as follows.
  - 1 mark per full answer / slide (notes complete, clear with any Q.s answered in full and correctly).
  - 1 mark for work submitted as a pdf file with a meaningful name formatted as follows.
    - Unit Number Section Meaningful Name Your Full English Name.
      - e.g.
        - 10 (a) Hello1 Questions Joe Bloggs
        - 10 (a) Hello1 Notes Joe Bloggs
- Final Course Grade-
  - Each Trimester @ 25%
  - Mock Exam/Final Exam/Independent Projects\* @ 25%
    - Final exam refers to the alternate exam for those who will not take an AP exam in May 2022.
      - The teacher will observe and evaluate your readiness for the AP exam and give his recommendation and communicate to the students.

Cumulative Course Grade (Prior to AP exam)

| AP | F  | D- | D  | D+ | C- | C  | C+ | B- | B  | B+ | A- | A |
|----|----|----|----|----|----|----|----|----|----|----|----|---|
| 5  | B+ | B+ | A- | A- | A- | A  | A  | A  | A  | A  | A  | A |
| 4  | C+ | C+ | B- | B  | B+ | A- | A- | A  | A  | A  | A  | A |
| 3  | C  | C  | C  | C+ | B- | B  | B+ | A- | A- | A  | A  | A |
| 2  | F  | D- | D  | D+ | C- | C- | C  | C+ | B- | B- | B  | B |
| 1  | F  | F  | F  | F  | D- | D  | D  | D  | D+ | C- | C  | C |

## INSTRUCTIONAL MATERIALS

The following instructional materials will be used for this course:

- [AP Computer Science A](#)
  - Textbooks will not generally be used but if you require some recommendations:
    - <https://www.skylit.com/jm.html>
    - <https://www.skylit.com/beprepared-java.html>
- You will need a laptop.

## CLASSROOM POLICIES

- Late work policy: 1 letter grade lower for each day late.
- You can only resubmit work under the following conditions:
  - You can ONLY resubmit once, and your 2nd score will be recorded as a new score (your original score will not be changed).
    - Therefore, your final score will be an average of your original & 2nd scores.
    - You must explain your original logic, why your answer is incorrect and why the correct answer is correct.
      - Please see:
        - [Submitting Corrected Code](#)
        - [Submitting Corrected MCQs](#)
- Positive Collaboration:
  - Showing and explaining your work (code/questions) to your peers.
  - Advising your peers.
  - Using and adapting code from the web, acknowledging its use and explaining in detail how it works.
- Plagiarism:
  - Sending your work (code/questions) to your peers.
  - Copying code solutions from the web without adaptation and passing it off as your own (not acknowledging its use) and/or not explaining in detail how it works.

- Units/Programs/Notes/Questions/Resources/Quizzes/Tests are planned out in the Teams [Calendar](#).
  - All work should be submitted in private chat in Teams (*not through use of assignment "hand in"*).
    - Programs (*from presentations & class/test FRQs*) should be submitted either in the form of a [Jdoodle](#) ([How to use Jdoodle](#)) shared link or a .java file.
    - Notes and answers to questions should be submitted as a pdf file with a fully explanatory name which ends with your full English name e.g. Data Types Questions Joe Blogs.
  - Quizzes & MCQ Tests:
    - Online MCQs which will automatically submit your answers to me and your email address.
- Homework:
  - I don't intend to formally set separate work as homework.
  - Generally homework will be to finish what you can't complete in class – see Teams [Calendar](#).

## COURSE OUTLINE

### [Defining your own Static Methods](#)

- Includes 'Passing by Value'.
- [Essential Knowledge & Learning Objectives](#)
- College Board Videos:
  - Static Variables and Methods:
    - [1](#), [2](#)
- FRQs:
  - [Combinatorics](#), [StringManip](#), [APCalendar](#), [CheckDigit](#), [DiverseArray](#), [LatinSquares](#)
- Code for the example 'Apartment' program:
  - Apt (*uses scanner class for user input*)
- [Practice-IT](#)
  - All remaining self checks & exercises.
- [CodingBat](#)
  - Create an account and enter my email address in "prefs" menu - "Teacher Share" section - "Share To" box and click "Share".
  - All problems except for "Recursion", "Functional" & "Map".
- [CodeStepByStep](#)
  - algorithms, arrays, console output (evenSumMax -> stanfordVsCal), expressions (all from expressionsExam1), if else, loops (all that were skipped in section [6](#)), parameters and return, searching, sorting, strings

### [Own Classes](#)

- [BlkAptTest.java](#)
- [input.txt](#)
  - (a) [Simple objects, Constructors & Overloading](#)

- [Hello1 Notes & Questions](#)
  - [Essential Knowledge & Learning Objectives](#)
  - [Hello1 Questions Only](#)
- [Hello2 Notes & Questions](#)
  - [Essential Knowledge & Learning Objectives](#)
  - [Hello2 Questions Only](#)
- [Hello3 Notes & Questions](#)
  - [Essential Knowledge & Learning Objectives](#)
  - [Hello3 Questions Only](#)
- [Hello4 Notes & Questions](#)
  - [Essential Knowledge & Learning Objectives](#)
  - [Hello4 Questions Only](#)
- AP College Board Videos:
  - Anatomy of a Class:
    - [1](#), [2](#), [3](#)
  - Constructors:
    - [1](#), [2](#), [3](#)
  - Documentation with Comments:
    - [1](#), [2](#)
  - Objects - Instances of Classes:
    - [1](#), [2](#)
  - Creating and Storing Objects (Instantiation):
    - [1](#), [2](#), [3](#)
  - Calling a Void Method:
    - [1](#), [2](#)
  - Calling a Void Method with Parameters:
    - [1](#), [2](#)
- [Practice-IT](#)
  - [BJP5 Chapter 8: Classes](#)
    - Self Checks 8.1 - 8.7
- [CodeStepByStep](#)
  - classes and objects
- (b) [Accessor & Mutator methods, Encapsulation, Overriding](#)
  - Contains Chair Static & Non-Static Combination Example
  - Square Root & Power Of
  - [Notes & All Questions](#)
    - [Essential Knowledge & Learning Objectives](#)
      - [Cone1 Questions Only](#)
      - [Cone2 Questions Only](#)
      - [Cone3 Questions Only](#)
      - [Bank Account 1 Questions Only](#)
      - [Bank Account 2 Questions Only](#)
  - AP College Board Videos:
    - Calling a Non-void Method:
      - [1](#), [2](#), [3](#)

- Accessor Methods:
  - [1](#), [2](#)
- Mutator Methods:
  - [1](#), [2](#)
- Writing Methods:
  - [1](#), [2](#), [3](#)
- Scope and Access:
  - [1](#), [2](#)
- this Keyword:
  - [1](#)
- Ethical and Social Implications of Computing Systems:
  - [1](#)
- [Practice-IT](#)
  - [BJP5 Chapter 8: Classes](#)
    - Self Checks 8.8 - 8.21, 8.23 - 8.31
    - Exercises 8.1 - 8.13
- (c) [Has-a](#)
  - [Forest Questions](#)
  - [Weather Report Questions](#)
  - [StringArray Questions](#)
  - [String\[\] args Questions](#)
  - [Practice-IT](#)
    - [BJP5 Chapter 8: Classes](#)
      - Exercises 8.14 - 8.22
- [FRQs:](#)
  - Write whole class:
    - [AdditionPattern](#), [CodeWordChecker](#), [StudyPractice](#), [HiddenWord](#), [StepTracker](#)
  - new obj creation:
    - [Thing](#), [LightSequence](#), [Successors](#)
  - objRef.method:
    - [Horse](#), [Kid](#), [NumberCube](#)
  - Constructor & methods:
    - [LightBoard](#), [Invitation](#), [UserName](#), [SkyView](#)
  - Static/Non-Static combination:
    - [PasswordGenerator](#), [Customer2](#)
  - Really Just Methods:
    - 1D arrays:
      - [SpinnerGame](#), [BatteryCharger](#), [Sound](#), [StudentRecord\(scores\)](#), [Trail](#)
    - 2D arrays:
      - [ExperimentalFarm](#), [GrayImage](#), [RouteCipher](#)
    - Other:
      - [TwoPlayerGame](#), [Vocab](#), [Robot](#), [Phrase](#)
- (d) [Inheritance](#)
  - [Learning Objectives & Essential Knowledge](#)

- [Notes & Questions](#)
  - [Questions Only](#)
- [FRQs:](#)
  - [Animal](#), [Item](#), [MenuItem](#), [Ticket](#)
- AP College Board Videos (*ignore any references to ArrayLists until we cover them later*):
  - Creating Superclasses and Subclasses:
    - [1](#), [2](#)
  - Writing Constructors for Subclasses:
    - [1](#), [2](#)
  - Overriding Methods:
    - [1](#), [2](#)
  - super Keyword:
    - [1](#), [2](#)
  - Creating References Using Inheritance Hierarchies:
    - [1](#), [2](#)
  - Polymorphism:
    - [1](#), [2](#), [3](#)
  - Object Superclass:
    - [1](#), [2](#)
  - [Creating Superclasses and Subclasses](#)
  - [Creating Superclasses and Subclasses](#)
  - [Overriding Methods](#)
- [InheritanceExamples](#)
- [SuperUseExamples](#)
- [AnotherPolymorphismExample](#)
- [PassingObjectsToMethodsRestrictionsExamples1](#)
- [Practice-IT](#)
  - [BJP5 Chapter 9: Inheritance and Interfaces](#)
- [CodeStepByStep](#)
  - inheritance
- [University of Washington CSE 143 \(CS2\)](#)
  - [CS2 Sections](#)
    - [Polymorphism](#)

### ArrayLists

- [Learning Objectives & Essential Knowledge](#)
- [Notes & Questions](#)
  - [Questions Only](#)
- AP College Board Videos:
  - Introduction to ArrayList:
    - [1](#), [2](#)
  - ArrayList Methods:
    - [1](#), [2](#), [3](#)
  - Traversing ArrayLists:
    - [1](#), [2](#), [3](#)

- Developing Algorithms Using Arrays:
  - [1](#)
- Searching:
  - [1](#), [2](#), [3](#)
- Sorting:
  - [1](#), [2](#), [3](#)
- Ethical Issues Around Data Collection:
  - [1](#)
- [Introduction to and Traversing ArrayLists](#)
- [ArrayList Methods](#)
- [Algorithms Using ArrayLists and Searching](#)
- [Algorithms Using ArrayLists \[Part 2\] and if-else Statements](#)
- [Algorithms Using ArrayLists \[Part 3\] and for/while Loops](#)
- FRQs
  - [CarRepair](#), [Appointment](#), [Climb](#), [CookieOrder](#), [Digits](#), [Flight](#), [FuelTank](#)
  - [Hotel](#), [LogMessage](#), [NumberTile](#), [scrambleWord](#), [SparseArray](#)
  - [StringFormatter](#), [StringPart](#), [StudentAnswerSheet](#)
  - [WordList](#), [WordPair](#), [Gizmo](#)
  - Inheritance & ArrayLists:
    - [Book](#), [Pet](#), [Pet2](#), [RandomStringChooser](#)
- [Practice-IT](#)
  - [BJP5 Chapter 10: ArrayLists](#)
- [CodeStepByStep](#)
  - collections (ArrayList)

## Recursion

- [Learning Objectives & Essential Knowledge](#)
- [Notes](#)
- [AP CS Recursive Tracing Lesson](#)
- [Merge Sort step by step walkthrough \(Recursion\)](#) (Video)
- [How to Code The Merge Sort Algorithm in Java](#) (Video)
- [MergeSortExample.java](#)
- AP College Board Videos:
  - Recursion:
    - [1](#), [2](#)
  - Recursive Searching and Sorting:
    - [1](#), [2](#), [3](#)
  - [Recursion](#)
  - [Recursive Searching and Sorting \[Part 1\]](#)
  - [Recursive Searching and Sorting \[Part 2\]](#)
- FRQs:
  - [NumberSystem](#), [ColorGrid](#), [Sentence](#)
- [Practice-IT](#)
  - [BJP5 Chapter 12: Recursion](#)
- [CodingBat](#)
  - [Recursion-1](#)

- [Recursion-2](#)
- [CodeStepByStep](#)
  - Recursion