

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

This elective course serves as a Java programming course that prepares students well for AP Computer Science courses. Students are introduced to the Java development cycle, object-orientation, and control structures. Students learn about primitive data types and their wrapper classes, operators, data structures such as arrays and lists. The course also emphasizes the problem-solving aspects of programming. Concepts covered in this course are further developed as students progress through the computer science discipline.

GRADING POLICY

General (CJ completion/Following School Rules/Collaboration/Participation/Proof Reading Resources)	10%
Minor: Classwork (Quizzes & Programs)	40%
Major: Tests (at the end of each unit)	50%

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.
- Final Course Grade-
 - Each Trimester @ 25%
 - Independent Projects @ 25%

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

1. Introduction to Java

2. How to Compile & Run your First Java Program

- [Error Explanations](#)
- College Board Videos:
 - [1](#), [2](#), [3](#)
- [Beginners Book](#)
- [Practice-IT](#) (create an account)
 - [Self-Check 1.7: outputSyntax](#)
 - [Self-Check 1.18: MyProgram-errors](#)
 - [Self Check 1.19: SecretMessage-errors](#)
 - [Self-Check 1.20: FamousSpeech-errors](#)
 - [Self-Check 1.21: methodHeaderSyntax](#)
 - [Exercise 1.6: Meta](#)
 - [CodeStepByStep](#)

- console output (HelloWorld)

3. Data Types & Identifiers/Variables in Java

- Includes 'Type Casting' from 1 primitive data type to another.
- [Learning Objectives & Essential Knowledge](#)
- [Notes](#)
- College Board Videos:
 - [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#)
- [Practice-IT](#)
 - [Building Java Programs, 5th edition - BJP4 Chapter 2: Primitive Data](#)
 - Self-Checks 1.6, 2.1 - 2.4 & 2.6 - 2.20 (*but not 2.5*)
 - [Exercise 2.1: displacement](#)
 - [CodeStepByStep](#)
 - console output: (Receipt), expressions: (digitExpressions & displacement)

4. Conversion

- [Exploding Dots Video](#)
- [Floating Point Video](#)
- [Number Conversion](#)
- [Conversion Test](#)
- [Practice-IT](#)
 - [1.1: binaryNumbers](#)
 - [1.2: binary1](#)
 - [1.3: binary](#)

5. Conditionals & Control Flow:

(a) if, if/else, if/else if

- [Learning Objectives & Essential Knowledge](#)
- [Notes](#)
- College Board Videos:
 - [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)

(b) Logical Operators

- Includes 'Short-Circuit Evaluation' & 'Debugging'.
- [Learning Objectives & Essential Knowledge](#)
- [Notes](#)
- College Board Videos:
 - [1](#), [2](#), [3](#), [4](#), [5](#)

• FRQs:

- [Menu](#), [PH](#), [APLine](#), [Item](#), [TheatreTickets](#)

• Practice-IT

- [Self-Check 4.2: logicExpressions1](#)
- [Self-Check 4.3: ifStatementSyntax](#)
- [Self-Check 4.10: javalsAwesome](#)
- [BJP5 Self-Check 5.14: logicExpressions2](#)
- [BJP5 Self-Check 5.22: deMorgans](#)

6. Loops:**(a) for**

- [Learning Objectives & Essential Knowledge](#)
- [Notes](#)
- College Board Videos:
 - [1](#), [2](#), [3](#), [4](#)
- [Practice-IT](#)
 - Self-Checks 2.21 - 2.25, [2.27](#), [2.28](#)
 - Exercise [2.2](#)

(b) nested for

- includes String Escape Sequences e.g. \", \\, \n, ...
- [Learning Objectives & Essential Knowledge](#)
- [Notes](#)
- College Board Videos:
 - [1](#)
- [Practice-IT](#)
 - Self-Checks 1.8 - 1.17, [2.26](#), 2.29 - 2.33, 2.36 - 2.38
 - Exercises 1.1 - 1.5 & [1.9](#), 2.4 - 2.17, 2.19 - 2.23
- [CodeStepByStep](#)
 - console output (FearTheTree & PrintProgram), loopMysteryPrint1 -> numberLoops4constant, starsPrint

(c) while

- Includes 'Debugging'.
- [Learning Objectives & Essential Knowledge](#)
- [Notes](#)
- College Board Videos:
 - [1](#), [2](#), [3](#), [4](#), [5](#)
- [Tracing Java Variables](#)
- [Practice-IT](#)
 - Self-Checks [5.1](#), [5.2](#), [5.13](#), [5.21](#)
- [CodeStepByStep](#)
 - loops (AsciiFigure)

• FRQ:

- [Digits](#)

7. Strings

- Integer.MAX_VALUE/MIN_VALUE & Double
- [Essential Knowledge & Learning Objectives](#)
- **FRQs:**
 - [Menu](#), [LongestStreak](#), [ScrambledWord](#), [LightSequence](#)
 - [StringManip](#), [Checker](#), [CodeWord](#), [HiddenWord](#)
- **Strings Class Quiz**
- AP College Board Videos:
 - String Objects - Concatenation, Literals, and More:
 - [1](#), [2](#), [3](#)
 - String Methods:

- [1](#), [2](#), [3](#)
- String Algorithms:
 - [1](#), [2](#), [3](#)
- Wrapper Classes - Integer and Double:
 - [1](#), [2](#), [3](#)
- Math Class:
 - [1](#), [2](#), [3](#)
- [String Objects: Concatenation, Literals, and More](#)
- [String Methods \[Part 1\]](#)
- [String Methods \[Part 2\]](#)
- [String Methods \[Part 3\] and Algorithms Using Strings](#)
- [String Methods \[Part 4\] and Comparing Objects](#)
- [Practice-IT](#)
 - Self-Checks 4.7 - 4.9, 4.11 - 4.14, [4.17](#), 5.23 - 5.26
- [CodeStepByStep](#)
 - console output (Receipt2-> CalculateLine)

8. Arrays:

(a) [1D](#)

- includes random(), abs
- [Essential Knowledge & Learning Objectives](#)
- **FRQs:**
 - Part FRQs:
 - [DivBySum](#), [Words](#), [ArraySum](#), [BatteryCharger](#), [Robot](#), [Scores](#), [Print Nums](#)
 - Full FRQs:
 - [Payroll](#), [NumberCube](#), [Sound](#), [TokenPass](#)
 - [Selection Sort Video](#)
 - [Insertion Sort Video](#)
 - [Merge Sort Video](#)
 - AP College Board Videos:
 - Array Creation and Access:
 - [1](#), [2](#), [3](#)
 - Traversing Arrays:
 - [1](#), [2](#), [3](#)
 - Enhanced for Loop for Arrays:
 - [1](#)
 - Developing Algorithms Using Arrays:
 - [1](#), [2](#), [3](#)
 - [Array Creation/Access and Math Class](#)
 - [Traversing Arrays](#)
 - [Enhanced for Loop for Arrays](#)
 - [Searching \[Part 1\]](#)
 - [Sorting \[Part 1\] \(Insertion Sort\)](#)
 - [Sorting \[Part 2\]](#)
 - [Practice-IT](#)

- [BJP5 Chapter 13: Searching and Sorting](#)

(b) 2D & final

- [Essential Knowledge & Learning Objectives](#)
- AP College Board Videos:
 - 2D Arrays:
 - [1](#), [2](#), [3](#)
 - Traversing 2D Arrays:
 - [1](#), [2](#), [3](#)
- **FRQs:**
 - **Part FRQs:**
 - [2d - 1d](#), [RouteCipher](#), [Crossword](#), [Successors](#)
 - **Full FRQs:**
 - [GrayImage](#), [LightBoard](#)
 - [Practice-IT](#)
 - [BJP5 Chapter 7: Arrays](#)
 - Self Checks 7.1 - 7.9, 7.12 - 7.15, 7.18, 7.22 - 7.28, 7.31 - 7.36