



# 2021 AI-Coding Online Courses for Children

## 1. Children 's AI Programming 000 for 5-7 years old

### **Scratch Jr, Introduction to the world of coding**

- Session #1: Animation, Stories and Games with Scratch Jr. – 15 activities

## 2. Children 's AI Programming 001 for 7-9 years old

### **Scratch Coding Based on Google CS First**

- Session #1: Storytelling with Scratch – 8 activities
- Session #2: Sports with Scratch – 8 activities
- Session #3: Game Design with Scratch – 8 activities
  - o Advanced topics and projects building in Scratch

## 3. Children 's AI Programming 002 for 9-12 years old

### **Introduction to Machine-learning based AI using Scratch**

- Session #1: Image Recognition, Natural Language Processing, Sentiment Analysis
- Session #2: Facial Recognition, Decision Trees, Speech Recognition
- Session #3: Brainstorming, Workshop, Create your own AI/ML project

## 4. Children 's AI Programming 003 for 10-14 years old

### **Getting started with Python programming**

- Session #1: Introductory Topics, Conditionals, Looping
- Session #2: Functions, Modules, Object-Oriented Programming

### **Python App and PyGame**

- Session #3: Design and Implementation of Games (and Applications)
  - o Introducing PyGame - 2-D Graphics Game Design using Python

## 5. Children 's AI Programming 005 for 11-15 years old

### **Introduction to Java programming**

- Session #1A: Introductory Topics, Data Types, System Output and Input
- Session #1B: Decision Structures and Boolean, Loops, Exceptions, Arrays

### **Advanced Java programming**

- Session #2A: Base Systems, Math Operators, and Object-Oriented Programming
- Session #2B: Threads, Advanced Exceptions, ArrayLists, New Additions

<https://school.thinkland.ai>

## Overview

Course ID	Course Name	Course Level			
AI000	Scratch Jr	Session #1			
AI001	Scratch	Session #1	Session #2	Session #3	Session #4
AI002	Scratch+AI	Session #1	Session #2	Session #3	
AI003	Python	Session #1	Session #2	Session #3	Session #4
AI005	Java	Session #1	Session #2		

\* Each session is 16 hours of class

## Class Selection Guideline

Student Age	Coding Experience	Course to Choose	Course We Recommend
5-6 y/o Kindergarten	None / Scratch Jr	000	000
7 y/o Grade 1	None / Scratch Jr	001	001
	Scratch	001*	
8-9 y/o Grade 2-3	None / Scratch Jr	001	001
	Scratch	001 / 002* / 003	002
	Python	002 / 003*	003
10-11 y/o Grade 4-5	None / Scratch Jr	001* / 002	001
	Scratch	001 / 002* / 003	002
	Python	002 / 003* / 005	003
12-14 y/o Grade 6-7-8 Middle School	None / Scratch Jr	001* / 003	001
	Scratch	003* / 005	003
	Python	003* / 005*	
	Java	003 / 005*	005
15 y/o Grade 9 High School	None / Scratch Jr	003	003
	Scratch	003 / 005*	
	Python	003 / 005*	005
	Java	003 / 005*	

# Children 's AI Programming 000

## Scratch Jr., Introduction to the world of coding

### AI000 Scratch Jr. COURSE OVERVIEW:

Scratch Jr is an introductory programming language that encourages creativity and expression, enabling five- to seven-year-old children to create their own interactive projects through coding. It provides young children with opportunities to design, create, and express themselves. Using Scratch Jr, children can use the programming blocks to bring their characters to life.

- **Getting Started:**

Students will learn the basics of using Scratch Jr. and make a dance party!

- **Animations – 4 activities:**

Students will create four small activities together, and then put all together to make a final animation.

- **Stories – 6 activities:**

Students will make their own stories by adding dialogue and new scenes to the animations. To make these stories come to life and play out like a movie!

- **Games – 5 activities:**

Students will learn how to make games. To make four different small games, and then use everything together to make one big game at the end.



## Children 's AI Programming 001

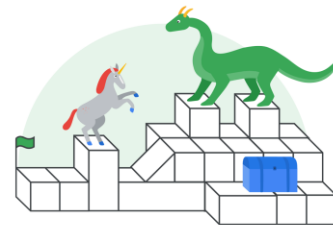
### Scratch 3.0 Coding Based on Google CS First

#### **AI001 Scratch COURSE OVERVIEW:**

Extending from Google's CS First themed curriculum, we introduce grades 2-5 students (7 years-old and up) to coding principles, all in project-based activities. Kids will systematically learn to program in Scratch, meanwhile explore storytelling techniques, build fun games and craft interesting presentations. The projects will showcase their creativity with the aid of coding and digital designs.

- **Session 1 (Storytelling with Scratch – 8 activities):**

Students use code to tell fun and interactive stories. Storytelling emphasizes creativity by encouraging students to tell a unique story each day.



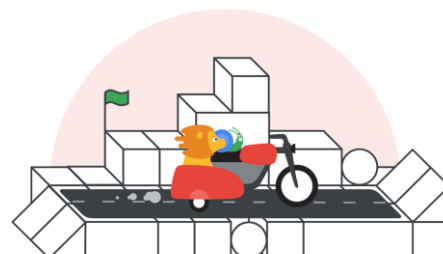
- **Session 2 (Sports with Scratch -- 8 activities):**

Students use computer science to simulate extreme sports, make their own commercials, and create commentary for a sporting event.



- **Session 3 (Game Design with Scratch – 8 activities):**

Students learn basic video game coding concepts by making different types of games, including racing, platform, and launching.



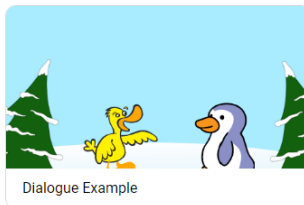
# Children 's AI Programming 001 Curriculum in Detail

## Scratch 3.0 Coding Based on Google CS First

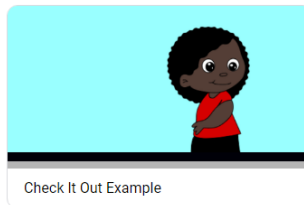
### AI001 Session#1: STORYTELLING (16 hours)



1. **Dialogue:** Learn about CS First and Scratch, then create a story in which two characters talk without using questions.
2. **Check It Out:** Tell a story in Scratch. In the story, a character walks through a scene describing what they see.
3. **Setting:** Create a dynamic stormy day setting, complete with rain and lightning.
4. **Premise:** Build a story around one of four premises.
5. **Characterization:** One character narrates another character's actions and thoughts.
6. **Interactive Storytelling:** Create a story in which the audience can make a decision.
7. **Personal Narrative:** Create a personal narrative based on one of three premises.
8. **Your Innovation Story:** Create an innovative product or idea, then use storytelling to pitch or sell it.



Dialogue Example



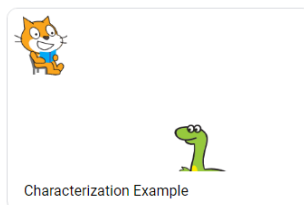
Check It Out Example



Setting Example



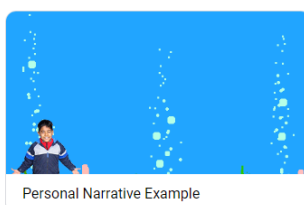
Premise Example



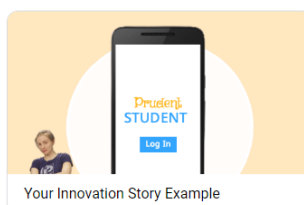
Characterization Example



Interactive Storytelling Example



Personal Narrative Example

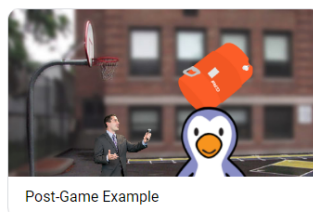
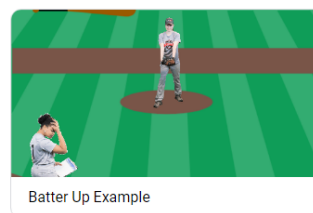
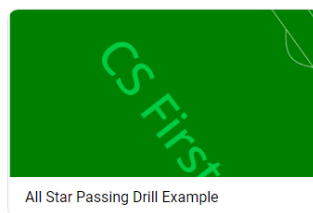
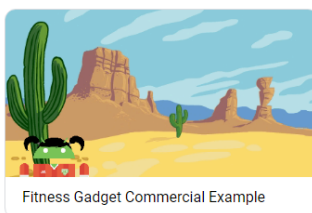
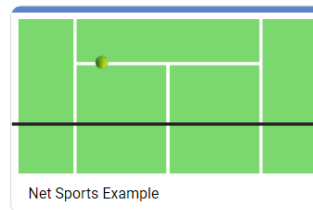
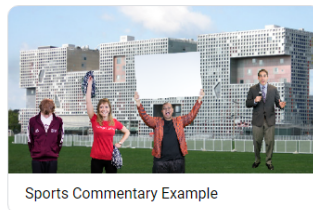
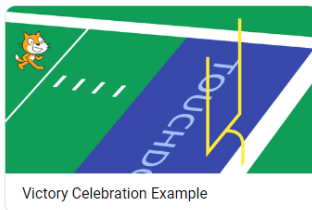


Your Innovation Story Example



## AI001 Session#2: SPORTS (16 hours)

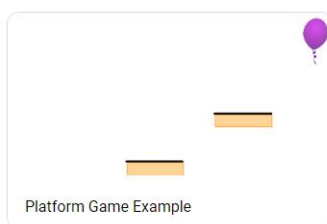
1. **Victory Celebration:** Learn about CS First and Scratch, then create a victory celebration in Scratch.
2. **Sports Commentary:** Create a dance, cheerleading, karate, or basketball performance competition. Then write the commentary for each of the competitions.
3. **Net Sports:** Create a fun net sport game with a bouncing ball and an athlete.
4. **Fitness Gadget Commercial:** Create a commercial around one of four fitness gadgets: wrist wear, eyewear, headwear, footwear.
5. **All-Star Passing Drill:** Build an all-star passing game where you try to aim at a moving receiver.
6. **Batter Up:** Build a home run derby style baseball game.
7. **Extreme Sports:** Build an extreme sports game in which a racer navigates an obstacle course for as long as possible.
8. **Post-Game Interview:** Build a project in which you get interviewed about what you programmed and learned in class.





## AI001 Session#3: GAME DESIGN (16 hours)

1. **Gaming Story:** Learn about CS First and Scratch, then create a gaming story.
2. **Racing Game:** Create a two-player racing game in which users control movement with the keyboard.
3. **Maze Game:** Create a game in which the user guides a sprite through a maze.
4. **Platform Game:** Create and learn about platform games. Students program a player sprite to move and jump across platforms.
5. **Escape Game:** Create an escape game in which a player must avoid other sprites that move randomly.
6. **Launcher Game:** Create a launcher game using key press events, clones, and variables.
7. **Quest Game:** Learn how to use storytelling in video game design while building an RPG style Quest Game.
8. **Cave Surfing Game:** Create a game with a side scrolling background (similar to the popular game Flappy Bird).





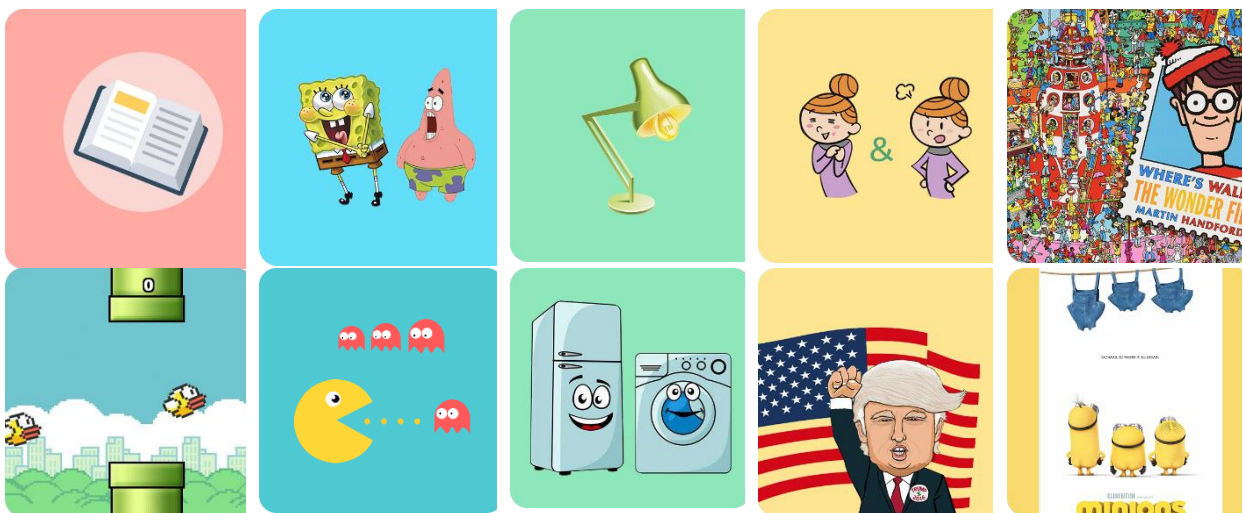
## Children 's AI Programming 002

### Introduction to Machine-learning based AI using Scratch

#### AI002 Scratch + AI Course Overview

Have you ever wondered how Alexa could always find an answer to your whimsical questions? We have and we figured that out, in Scratch. Using the proprietary AI platform built by Thinkland.ai, we introduce grades 4-8 students to the concept of machine learning and its applications in real life. Learning with 10 projects from different AI focuses all implemented in Scratch, the kids will explore the meanings of artificial intelligence in our daily life and the engineering principles behind these AI-based applications.

- **Session 1:**
  1. Introduction to Machine Learning
  2. Image Recognition & Spongebob and Friends
  3. Natural Language Processing & Simple Smart Home
  4. Sentiment Analysis & Praises and Criticisms
- **Session 2:**
  5. Facial Recognition & Facelock
  6. Decision Trees & Flappy Bird
  7. Decision Trees & Pacman
  8. Speech + Speaker Recognition & Voice Lock
- **Session 3**
  9. Brainstorming Ideas + Workshop
  10. Create your own project & Implement Multiple Machine Learning Concepts





Children 's AI Programming 002 in Detail  
**Introduction to Machine-learning based AI using Scratch**

**AI002 Session#1 (16 hours)**

1. INTRODUCTION TO MACHINE LEARNING
  - a. Introduce concepts of Artificial Intelligence and Machine Learning, Strong AI vs Weak AI to students. Discuss what a machine learning model is. Also briefly discuss future topics. Includes a project that teaches computers the ability to identify an apple from and orange.
  - b. Project: Apple or Orange. Create a Machine Learning model that can allow the computer to identify an apple from and orange.
2. IMAGE RECOGNITION & SPONGEBOB AND FRIENDS
  - a. Discuss image recognition, including how it works, applications. Also create an image recognition model that helps the computer to identify images and create a scratch project that utilizes this model.
  - b. Project: Spongebob and Friends. Help the computer send different images of SpongeBob, Patrick and other characters to their home/destination using a machine learning model.
3. NATURAL LANGUAGE PROCESSING AND SIMPLE SMART HOME
  - a. Discuss Natural Language Processing, how it works, as well as applications of it in the real world. Create a project that utilizes the NLP learning model.
  - b. Project: Smart Home (Simple Version). Create a project that allows the player to type commands of any variation to open and close home appliances.
4. SENTIMENT ANALYSIS & PRAISES AND CRITICISMS
  - a. Discuss sentiment analysis, how to teach the computer to recognize sentiments, as well as applications of this concept. Create a project that utilizes the corresponding machine learning model.
  - b. Project: Praises or Criticisms. In Scratch, create a program that reacts accordingly to the user's comments; if the comment is positive or is a praise, the program will react happily. If the comment is negative or a criticism, the program will be sad.

## **AI002 Session#2 (16 hours)**

### **5. FACIAL RECOGNITION & FACELOCK**

- a. Discuss facial recognition, how a computer can recognize and identify a person through facial features, as well as applications. Create a project that utilizes the corresponding machine learning model.
- b. Project: Facelock. Create a simulation of a phone that will unlock depending on the user's face, similar to an iPhone X.

### **6. DECISION TREES & FLAPPY BIRD**

- a. Discuss decision trees, an machine learning algorithm, and how computers can use this concept to make decisions. Create a flappy bird project that uses this machine learning algorithm.
- b. Project: Flappy Bird. Create a flappy bird game through Scratch, in which the computer will learn and adapt a human player's strategy, and be able to play flappy bird by itself.

### **7. DECISION TREES & PACMAN**

- a. Continue the discussion on decision trees and create another scratch program that uses decision trees to allow the computer to play a game.
- b. Project: Pac-Man. Create a Pac-Man game through Scratch, in which the computer will learn from the human player to play a simplified version of Pac-Man.

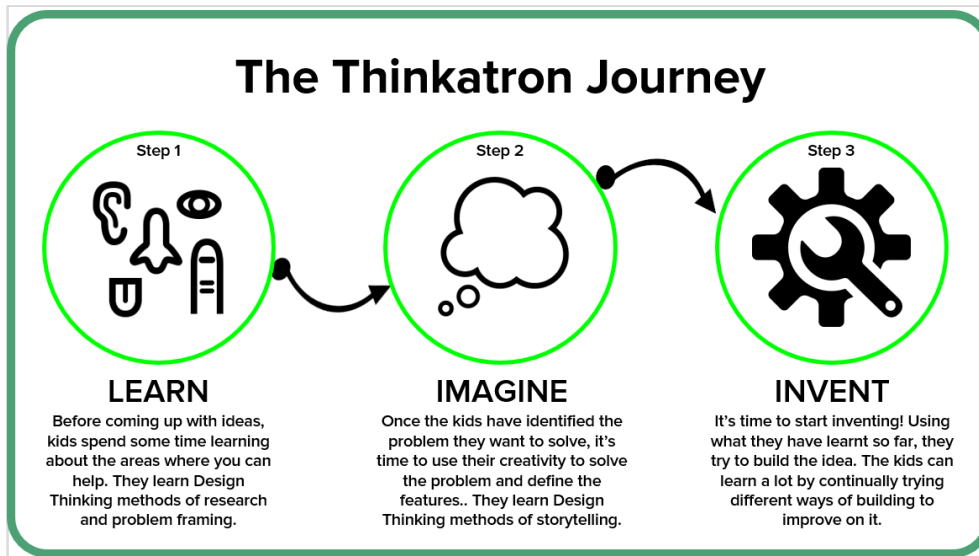
### **8. SPEECH+SPEAKER RECOGNITION & VOICE LOCK**

- a. Discuss speech and speaker recognition, their differences, and their applications. Further discuss the idea of two factor authentication. Create a scratch project that utilizes both machine learning concepts.
- b. Project: Voice Lock. Create a simulation of a phone in which it can be unlocked if the speaker's voice matches the owner's voice. Further enhance the project by implementing two factor authentications, which forces the user to say a randomized specific phrase in order to bypass the security.

## AI002 Session#3 (16 hours)

9. Brainstorming Ideas + Workshop

10. Create your own project & Implement Multiple Machine Learning Concepts



# Children 's AI Programming 003

## Getting started with Python programming

### AI003 Python Course Overview:

An introductory course for those interested in learning Python, programming, and how to use programming in various mathematical and scientific tasks. Students will install Python 3.8, learn the basics of Python programming, and apply the basics to various tasks. And then gradually expand to computational thinking skills behind software application design.

### Introduction to Python

#### Session 1:

1. Introductory Topics
2. Conditionals
3. Looping
4. Fun projects!

#### Session 2:

5. Functions
6. Modules
7. Object-Oriented Programming

### Python App and PyGame

#### Session 3:

8. Part I : Python Review
9. Part II : Design and Implementation of Games (and Applications)
10. Part III : Introducing PyGame - 2-D Graphics Game Design using Python



## Children 's AI Programming 003 in Detail

### **Getting started with Python programming**

#### **AI003 Session#1 (16 hours)**

1. Introductory Topics
  - Basics of Numbers
  - Basics of Strings
  - Taking input/output
  - Variables, arrays, and tuples
  - Program Flow
2. Conditionals
  - Booleans and comparators
  - If-elif-else statements
  - And/or/not
  - Nesting
3. Looping
  - While loops
  - For loops
  - For loops v2
  - Break, continue
4. Fun projects!
  - Drawing with ``turtle``
  - Statistics with ``numpy``

#### **AI003 Session#2 (16 hours)**

5. Functions
  - Desc: Of the math kind and not of the math kind.
6. Modules
  - Desc: Building projects in pieces.
7. Object-Oriented Programming
  - Desc: Classic.

## **AI003 Python App and PyGame:**

### **AI003 Session#3 (16+ hours)**

This session aims to reinforce the python basics in a project based curriculum while gradually expanding to introduce computational thinking skills behind software application design. Some previous basics will be reviewed, reintroduced and reinforced, and new topics will be introduced along the way. The project based approach employs a pedagogical sequence: project demo -> requirements analysis -> problem presentation -> solution proposal and implementation -> reinforcement of syntax and coding concepts.

### **Part I - Python Review**

This part aims to review the basics of Python that may have been taught in the prior sessions.

1. Project 1 - Guess the Number (1 - 1.5 hour)
  - Topics: import • randint() • for • Blocks • str(), int(), float() • Booleans • Comparison • Conditional
2. Project 2 - A Joke-Telling Program (1 hour)
  - Topics: Escape • ' and " • print()
3. Project 3 - Dragon Realm (1 - 2 hour)
  - Topics: functions • Multiline strings • while • Boolean operators • return • variable scope • Parameters and arguments • sleep()

### **Part II - Design and Implementation of Games (and Applications)**

This part uses projects of increasing complexity to illustrate the computational thinking behind the design and implementation (including testing) of software applications while reinforcing previously acquired Python basics.

1. Interlude - Using the Debugger (1 hour)
  - Topics: 3 types of errors • debugger • Go and Quit • Stepping into, over, and out • Breakpoints
2. Project 4a - Designing Hangman with Flowcharts (1.5 - 2 hour)
  - Topics: ASCII art • Designing with flowcharts
3. Project 4b - Writing the Hangman Code (2 - 3 hours)
  - Topics: Lists • in operator • Methods • split(), lower(), upper(), startswith(), and endswith() • elif
4. Project 4c - Extending Hangman (2 hours)
  - Topics: dict • Key-value pairs • keys() and values() • multi-var assignment
5. Project 5 - Tic-Tac-Toe (3 - 4 hours)
  - Topics: AI • List references • Short-circuit • None
6. Project 6 - The Bagels Deduction Game (2 – 3 hours)
  - Topics: random.shuffle() • Augmented assignment operators • sort() • join() • String interpolation • conversion specifier %s • Nested loops
7. Interlude - The Cartesian Coordinate System (1 hour)

- Topics: x-axis and y-axis • Negative numbers • Pixels • commutativity • abs()
- 8. Project 7 - Sonar Treasure Hunt (4 hours)
  - Topics: Data structures • Pythagorean theorem • remove() • isdigit() • sys.exit()
- 9. Project 8 - Caesar Cipher (2 – 3 hours)
  - Topics: Cryptography and ciphers • Ciphertext, plaintext, keys, symbols • Encrypt/decrypt • Caesar cipher • find() • Cryptanalysis • The brute-force
- 10. Project 9a - The Reversegam Game (4 – 5 hours)
  - Topics: bool() • Simulating moves on a Reversegam board
- 11. Project 9b - Reversegam AI Simulation (2 – 3 hours)
  - Topics: Simulations • Integer division • round() • Computer-versus-computer games

## Part III - Introducing PyGame - 2-D Graphics Game Design using Python

This part introduces the basics of a popular Python library (pygame) as a bridge to the concepts common in computer game development while further developing the students' software development skills.

1. Project 10 - Creating Graphics (1.5 – 2 hours)
  - Topics: pygame • Colors and fonts • Aliased and anti-aliased graphics • Attributes • Font, Surface, Rect, PixelArray • Constructor functions • drawing functions • The blit() method for surface objects • Events
2. Project 11 - Animating Graphics (1.5 – 2 hours)
  - Topics: Animating objects with the game loop • Changing the direction of an object
3. Project 12 - Collision Detection (1.5 – 3 hours)
  - Topics: Clock objects • Keyboard input in pygame • Mouse input • Collision • list iteration caveat
4. Project 13 - Using Sounds And Images (1.5 – 3 hours)
  - Topics: Sound and image • Drawing and rescaling sprites • Adding music and sounds • Toggling sound on and off
5. Project 14 - A Dodger Game with Sounds and Images (1.5 – 3 hours)
  - Topics: pygame.FULLSCREEN • move\_ip() Rect method



## Children 's AI Programming 005

### **Introduction to Java and Advanced Java**

#### **AI005 Java Course Overview:**

An in-depth course detailing everything related to java, from basic concepts of how a computer functions to advanced concepts that appear on the AP exams. This course blends rigor with fun as students participate in hands-on projects and labs to better understand the language of java and take the step into more advanced coding.

#### **Introduction to Java**

##### **Session 1:**

1. Introductory Topics
2. Data Types
3. Looping
4. Project Lab 1 - Mini market

##### **Session 2:**

5. Advanced Boolean and Decision Structure
6. Loops and Iterators
7. Project Lab 2 - Rock paper scissors game
8. Basics of Errors and Exception
9. Array and ArrayLists
10. Final Project - Mini market revisited

#### **Advanced Java**

##### **Session 3:**

1. Binary, hexadecimal, and ASCII
2. Math class and arithmetic functions
3. Project lab 3 – Create a calculator
4. Object-oriented concepts part 1
5. Object-oriented concepts part 2
6. Project lab 4 - Create your own objects

##### **Session 4**

7. Threads, Runnables and Concurrency
8. Advanced Exceptions and Errors
9. Advanced ArrayLists and list
10. Lambda Functions and new java additions
11. Final project - Create your own objects expanded

## Children 's AI Programming 005 in Detail

### **Introduction to Java and Advanced Java**

#### **AI005 Introduction to Java:**

##### **Session#1A (6 hours)**

1. Introductory Topics
  - CPU, GPU, SSD/hard drive, and RAM
  - Memory allocation
  - CPU cycles
  - Strategic programming
  - Java formatting
  - Java keywords
2. Data Types
  - Primitive data types
  - Strings and StringBuilder objects
3. Looping
  - Scanner objects
  - Data collection
  - Outputting formatted information
4. Project Lab 1
  - Mini market: Students will create a project that prompts the user for purchases and can output a receipt following a format.

##### **Session #1B (10 hours)**

5. Advanced Boolean and Decision Structure
  - Boolean operators
  - If and if-else statements
  - Ternary operators
  - Switch statements
6. Loops and Iterators
  - For, while, and do-while loops
  - Nested loops
7. Project Lab 2
  - Rock paper scissors game: students will create a program that can play rock paper scissors against the user
8. Basics of Errors and Exception
  - Compile-time errors
  - Run-time errors

- Handling and try-catch blocks
- Outputting formatted information
- 9. Array and ArrayLists
  - Arrays and ArrayLists objects
  - How to navigate and find things in arrays
- 10. Final Project
  - Mini market revisited: Students will build onto their first project lab using arrays and loops to make the program more efficient and detailed.

## **AI005 Advanced Java:**

### **Session#2A (11 hours)**

1. Binary, hexadecimal, and ASCII
  - Converting between the different base systems
2. Math class and arithmetic functions
  - Converting and casting between different casts
  - All arithmetic methods declared in the javadocs
3. Project lab 3
  - Students will create a calculator using the math functions that solve a mathematical problem
4. Object-oriented concepts part 1
  - Accessor and constructor methods
  - Inheritance and polymorphism
5. Object-oriented concepts part 2
  - Interfaces
  - Enums
  - Abstract classes
6. Project lab 4:
  - Create your own objects: students will use objects to create their own network of code

### **Session #2B (11 hours)**

7. Threads, Runnables and Concurrency
  - Threads, Runnables, and concurrency
8. Advanced Exceptions and Errors
  - Custom exception
  - Throw and finally code blocks
9. Advanced ArrayLists and list
  - Lists, LinkedLists, and other collections

## 10. Lambda Functions and new java additions

- New additions to java

## 11. Final project:

- Create your own objects expanded
- Use everything you learned to create anything you want



Email us : [contact@thinkland.ai](mailto:contact@thinkland.ai)