https://technokids.com/store/collections/coding-for-kids-activities

TECHNOKids®

CODING

Curriculum Collection

PROJECT-BASED COURSE: COMPUTER SCIENCE AND AI LESSONS

A collection of technology courses for
Scratch, Python, HTML, and more!

Contents

Coding Collection Courses and Software 2
Coding Collection Course Descriptions: Primary (Grades 1-3) 3
TechnoWhiz 3
TechnoTales 4
Coding Collection Project Descriptions: Junior (Grades 3-6) 5
TechnoArcade 5
TechnoRace 6
TechnoTurtle 7
Coding Collection Course Descriptions: Intermediate (Grades 6-9) 8
TechnoBot AI 8
TechnoCode 9
TechnoHTML5 10
TechnoPython 11
Coding Collection Course Descriptions: Senior (Grades 8-12) 12
TechnoChatbot AI 12
TechnoFuture AI 13
Contact Information 14

Coding Collection Courses and Software

TechnoKids Technology Courses are available for a variety of apps and programming languages. Refer to the table to identify courses that are right for you!

Suggested grade levels:	Scratch Jr	Scratch	Python 3	Generative AI	Text Editor	Bot Libre
Primary | Grades 1-3						
TechnoWhiz Become a programming 'whiz kid'. Build simple scripts and loops to create silly scenes, feed a pet monster, explore a magical land, and invent a racing game.	•					
TechnoTales Blend coding with storytelling. Design a modern fairy tale that has a hero go on a quest. Build scripts to animate the story action.	•					
Junior | Grades 3-6						
TechnoArcade Design arcade games. Build Jumble Tumble, Let's Jam, Mystery Island, and Lost Treasure. Invite friends to an online arcade.		•				
TechnoRace Develop an original game. Players race to complete a mission before time is up. To win they must avoid obstacles and collect treasure. Collaborate to test game design.		•				
TechnoTurtle Develop and debug code to conquer mazes, paint pixel art, create a Mad Lib Generator and build a carnival game.			•			
Intermediate | Grades 6-9						
TechnoCode Spark an interest in computer science. Design an Activity Studio for kids using Scratch. Build blocks of code to design animations, puzzles, stories, and games.		•				
TechnoBot AI Program a drone delivery system, robot pick-up service, and self-driving tour that solve real world problems using artificial intelligence.		•				
TechnoPython Program a series of games using Python including Pet Monster Rescue, Guess It, and Adventure Quest. Share your favorite one in a coding presentation			•			
TechnoHTML 5 Develop a web page using HTML and CSS. Write code to set the style of the background, text, lists, graphics, hyperlinks, and tables. Upload to the Internet.					•	
Senior | Grades 9-12						
TechnoFuture AI Research emerging and futuristic AI technologies. Use Generative AI to create images, models, and music for a sci-fi story that imagines the future. Practice digital ethics.		•		•		
TechnoChatBot AI Become a chatbot developer. Improve the lives of others. Build an event chatbot, order chatbot, and virtual agent.		•				•

Coding Collection Course Descriptions: Primary (Grades 1-3)

TechnoWhiz

In this course, students jump into the world of coding. They learn how to sequence blocks in Scratch Jr to build simple scripts and loops. The coders design silly scenes, feed a pet monster, explore a magical land, race to the finish line, and more! This fun-filled introduction to programming will spark students' imaginations. Invite curious young minds to become whiz kids!

The technology course has 16 assignments that are divided into 6 Sessions:

- Session 1 Become a Whiz Kid
  In session 1, students become programming whiz kids. Their first assignment is to sequence a series of tasks to learn about the job of a programmer. Next, they open Scratch Jr to investigate the function of many of the tools. By adding Motion blocks to the programming area, they learn how to code a character's movement. Next, they connect blocks together to form a script that sequences actions. Finally, students cause the script to loop. At the end of the session, they combine their coding skills to create a silly scene. It comes to life by combining a colorful background with moving characters.
- Session 2 Count and Code
  In session 2, the programming whiz kids take the Counting Challenge. They learn to direct how many steps a character moves to reach a goal. To start, they solve puzzles to help the cat eat the cake, the dog catch the ball, and the horse reach the barn. Next, they explore how to edit the Motion coding blocks to set the number of steps. Once they have mastered this skill, students build scripts to help characters get home. Can they do it?
- Session 3 Create a Pet Monster Game
  In session 3, the programming whiz kids design a game. They build code that has a player feed a pet monster. To start, they create an opening sequence to attract attention. It has the character talk, grow, shrink, and blink. The action is controlled using the Wait and Repeat blocks. Afterwards, students apply their knowledge to add food to the stage that when clicked moves to the pet monster and then disappears. What does the pet monster like to eat?
- Session 4 Design a Magical Land
  In session 4, the programming whiz kids design an interactive magical land. The game invites players to explore the place to discover surprises. By clicking on objects, they can cause flowers to grow, a sun to spin, or dragons to hop. The speed of the action is controlled to make it happen slow or fast. What amazing things are hidden in the scene?
- Session 5 Race to the Finish
  In session 5, the programming whiz kids build a racing game. Players watch the action and then must select the winner. If they pick correctly, the racer will do a celebration dance. Who will reach the finish line first?
- Session 6 Game Time
  In session 6, the programming whiz kids invite others to play their racing game. Players rate what they liked the most about the activity. Afterwards, the young game designers answer questions about coding the Scratch Jr project.

Extension Activities:
Move a Character Diagonally, Move Around Town Game, Be a Gamer, Edit a Character, Paint a Background, I am a Programmer

Technology Skills: programming
Technology Integration: Computer Science, Mathematics, Language Arts, Social Studies, Visual Arts
Software Applications: Scratch Jr

TechnoTales

In this course, students make a modern fairy tale. Using Scratch Jr, they will combine coding blocks to form scripts that animate the story action. The tale will be about a character that overcomes a problem by going on a quest. To live happily ever after, they must find a hidden item and locate someone that can help. What will happen in the "Once Upon a Time" adventure?

The technology course has 21 assignments that are divided into 6 Sessions:

- Session 1 Get to Know Scratch Jr
  In session 1, students are introduced to programming with Scratch Jr. To start, they explore the program window to learn about commonly used tools. Afterwards, they investigate the Triggering, Motion, Looks, Sound, Control, and End blocks to discover their function. Once familiar with the coding blocks they build simple scripts to create an animated scene.
- Session 2 Once Upon a Time
  In session 2, students begin to create their Techno Tale. Their modern fairy tale will be told by building scripts using coding blocks in Scratch Jr. To gain inspiration they watch an example story and answer questions about the setting, characters, and plot. Next, they complete a planning sheet to organize their ideas. Students then design their first story page. It introduces the problem and shows the hero embarking on a quest to find a solution. Motion and Looks blocks are used to animate the action.
- Session 3 Embark on a Quest
  In session 3, students design another page in their Techno Tale. In this part of the story, the hero embarks on a quest. They travel in search of a special item. The reader will join in the hunt by tapping objects on the page. When the correct location is found, the item will reveal itself. Wait and Repeat blocks are used to control the timing of each action.
- Session 4 Seek Help
  In session 4, students create the third page in their Techno Tale. In this part of the story, the hero seeks help. If-then logic is used to control the animation. Scripts start only when characters bump into one another. This produces a fun sequence of events. First the hero asks for help. Then the helper goes after the villain. Finally, the bad guy does an action to show the problem is solved. To prepare to create this story page, students learn how to display the grid and count steps to direct movement.
- Session 5 Happily Ever After
  In session 5, students animate their final page in their Techno Tale. At the ending of the tale the characters live happily ever after. Broadcasting is used to organize the timing of events. This form of conditional logic directs scripts to start only when a message is received. It is used to trigger characters to do an action to celebrate the problem being solved.
- Session 6 Story Time
  In session 6, students share their Techno Tale. To prepare the project for viewers, a checklist is used to guide revisions. The story is then shown to friends and family. At the end of TechnoTale, the young coders reflect upon the learning experience.

Extension Activities:
Record a Sound, How to Design a Character, Set the Speed, Explore the Kingdom, Design Your Own Quest, Animate with Coded Messages, Connect the Pages

Technology Skills: Programming
Technology Integration: Computer Science, Language Arts, Mathematics, Social Studies, Visual Arts
Software Applications: Scratch Jr

Coding Collection Project Descriptions: Junior (Grades 3-6)

TechnoArcade

In this course, students become game developers. They use Scratch coding blocks to create activities for kids. These include Jumble Tumble, Let's Jam, Mystery Island, and Lost Treasure. Upon completion, gaming fans visit an online arcade to share in the fun.

The technology course has 18 assignments that are divided into 6 Sessions:

- Session 1 Arcade Hero
  In session 1, students become arcade heroes. They prepare to build an online arcade for kids. To start, they register for a free Scratch account to gain access to the online coding platform. Afterwards, they explore the program to learn about common tools and terminology. By connecting blocks to make a script they discover how to control a sprite.
- Session 2 Jumble Tumble
  In session 2, students build Jumble Tumble. In this game a press of a key creates a mish mash of characters dashing and rolling across the screen. This chaotic scene uses Motion blocks to control movement. By sequencing the coding blocks, students discover how to direct and loop action. For an extra challenge, they can use if-then logic to create a silly outcome when two sprites crash.
- Session 3 Let's Jam
  In session 3, students put together a band. They design a game that transforms the keyboard into a musical instrument. Players will enjoy leading a jam session as they tap away at the keys. Students wanting a personal touch can add a variable that allows the player to name the new music group.
- Session 4 Mystery Island
  In session 4, students create Mystery Island. In this game, players explore an imaginary land. By tapping objects, they discover strange creatures. To build the code, Looks blocks are combined to change the appearance of sprites. Coders wanting to create an extra surprise can use random operators.
- Session 5 Lost Treasure Part One
  In session 5, students begin to create the game Lost Treasure. In it, players collect items to score points. The player moves the mouse to pick up lost items and carries them to a specific spot. To start, students plan the theme of their game. Next, they insert a setting, hero, and treasure. With this complete, the game designers build code to control the movement of the hero. Sounds and Looks blocks combine to add excitement to the game play.
- Session 6 Lost Treasure Part Two
  In session 6, students finish designing the game Lost Treasure. They create a variable to keep score. Using the wait until coding block, they build a script that ends the game when all hidden items are found. An optional activity has students create a timer to have players race against the clock. Upon completion, a tester plays the game.

Extension Activities:
Explore the Scratch Community, Dash and Crash, Pick a Band Name, Surprise Me, Add a Game Title, Create a Timer

Technology Skills: Programming
Technology Integration: Computer Science, Mathematics
Software Applications: Scratch

TechnoRace

In this course, students become game developers. They build an imaginary world using Scratch coding blocks. This online rescue mission has players race against time to collect points. Loops, conditionals, and variables combine to produce original game play. Upon completion, gaming fans test the story action. For coders wanting an extra challenge, they can customize animation, create flashing backdrops, or increase difficulty level.

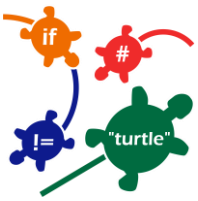
The technology course has 21 assignments that are divided into 6 Sessions:

- Session 1 Start from Scratch
  In session 1, students are introduced to Scratch, an online coding platform. To start, they explore the program to learn about common tools and terminology. Once familiar with the programming environment, students compete in a racing adventure. Afterwards, they experiment with the code to alter the player's experience. This exploration provides a foundation for building their own game.
- Session 2 Become a Game Developer
  In session 2, students become game developers. They invent a storyline for a rescue mission. In it, the player races against time to reach a goal. Along the way they must collect treasure and avoid obstacles. Once students have a plan, they begin to build the game board. First, they insert sprites onto the stage to act as the player, treasure, obstacle, and goal. Next, they use the Paint Editor to create an imaginary world. Afterwards, they build a simple script that will play theme music throughout the game.
- Session 3 Let's Get Moving
  In session 3, students create game controls. The fun starts with an exploration of Motion blocks. They build a script that moves the goal sprite, so it attracts attention. Next, the game developers transfer their knowledge to build controls using the arrow keys. Once the player can move around the imaginary world, students learn how to use logic to prevent walking through objects. Students in need of a challenge can build a script that teleports or launches the player at hyper speed. It is time to get moving!
- Session 4 Avoid Obstacles to Win
  In session 4, students test the player's skill by restricting their movements. To begin, they code a looping script that moves a sprite on the stage so that it temporarily blocks the player. Next, the game developers design code that causes the player to slow down if it touches the obstacle. With this script complete, students apply their knowledge to stop the game when the player reaches its goal. For those wanting to add even more interest, they can switch backgrounds when two sprites collide, or the game is over.
- Session 5 Collect Treasure
  In session 5, students design scripts that allow the player to collect points. The first task is an exploration of the Looks blocks to change the appearance of treasure, so players take notice. Next, students learn about variables. They apply this knowledge to calculate points when a player touches an object such as a coin or jewel. To enhance the game, students can elect to use the Paint Editor to customize the animation of a sprite.
- Session 6 Time is Up
  In session 6, students complete the game by adding a timer to increase the difficulty level. They apply their knowledge of variables to build scripts that track time. When a limit is met the game ends. Upon completion, students invite others to test the story action and provide feedback. It is going to be a race to the finish!

Extension Activities:
Digital Footprints in Scratch, Upload a Sprite, Jump to Hyper Speed, Flash a Backdrop, Paint a New Costume, Add Comments

Technology Skills: Programming
Technology Integration: Computer Science, Mathematics
Software Applications: Scratch

TechnoTurtle



In this course, students become game designers. They use Python and the Turtle library to conquer mazes, paint pixel art, create a Mad Lib Generator, and build a Carnival Game. The fun begins when students edit code to gain an understanding of the structure of Python scripts. Once familiar with basic concepts, the young programmers are introduced to debugging, loops, variables, and conditional logic. Ignite an interest in programming with meaningful activities designed for beginners.

The technology course has 30 assignments that are divided into 6 Sessions:

- Session 1 Python, Turtles and Bugs
  In this session, students become programmers. To start they learn how the Python programming language is used in daily life. Next, they visit the Turtle library to study the commands and make predictions about their function. They test their ideas by modifying a program to control what it draws. Once familiar with running a Python program, students add bugs to the code in order to identify and fix common errors.
- Session 2 Conquer the Maze
  In this session, students control the movement of a Turtle through a series of mazes. The fun begins when the young programmers write their first script. It marches a Turtle around the screen by moving forwards, backwards, and turning. Once they have mastered this set of commands, students are challenged to develop a script that will guide a Turtle through a maze. Can they solve the puzzle?
- Session 3 Draw Pictures
  In this session, students write code to draw pictures. To start, they learn how to plot a point on the canvas using x and y coordinates. They apply this knowledge to stamp a unique design. Next, the young programmers follow instructions to design a robot by combining lines, rectangles, circles, dots, and symbols. Once familiar with how to control the Turtle's drawing tools, students build their own program to draw a picture.
- Session 4 Design Colorful Spirographs
  In this session, students paint stunning artwork. To start, they learn code that repeats a set of instructions forever or for a specific number of times. Next, they complete a series of exercises to discover how to construct looping geometric shapes called spirographs. Once students are familiar with designing patterns, they use the Random library to produce colorful creations.
- Session 5 Create a Mad Lib Generator
  In this session, students design a word game, called a Mad Lib. It has players provide a list of words that are used to complete a silly sentence or story. To prepare for this coding task, students learn about variables by chatting with the computer. Next, they edit a Mad Lib party invitation to discover how to join variables and text together to form sentences. Once familiar with the structure of the code, they program their own wacky word game.
- Session 6 Invent a Carnival Game
  In this session, students become game designers. They combine Python and Turtle programming commands to produce a Carnival Game. To start, they learn about if, elif, and else. Once familiar with conditional logic they invent a game that prompts the player to pick an option to win a prize. Optional challenges enrich the design such as looping a flashing message or showing a picture of their winnings. Get ready for fun. Step right up to win a prize!

Extension Activities:
Imagine Life Without Coding, Dot-to-Dot Fun, Customize the Stamp, Rain Cats and Dogs, Build a Word Game, Guess a Number

Technology Skills: Programming
Technology Integration: Math, Language Arts, Visual Arts, Social Studies
Software Applications: IDLE Python 3

Coding Collection Course Descriptions: Intermediate (Grades 6-9)

TechnoBot AI

In this course, students become artificial intelligence specialists. They apply a design thinking model to imagine creative solutions to real-world problems. Using Scratch, they build prototypes of their inventions. These include a drone delivery system, robot pickup service, and self-driving tour. Afterwards, they present one of their AI prototypes as an investment opportunity. Throughout the project, they reflect upon the possibilities and limitations of AI technologies.

The technology course has 24 assignments that are divided into 6 Sessions:

- Session 1 Introduction to AI
  In this session, students become artificial intelligence specialists. This role requires them to solve problems using AI and Scratch. The fun begins with an exploration of AI in daily lives. Next, they register for a Scratch account and discover how to use coding blocks to create a simple animation. They will apply this knowledge in upcoming sessions to program a drone delivery system, robot pickup service, and self-driving tour.
- Session 2 Flying Machines at School
  In this session, students develop a prototype of a drone delivery system that uses AI. It must solve a common problem at school. Using Scratch, students build a simple program that flies a sprite-drone to collect items and then return to its original start point. They will refine the code to adjust to new delivery locations and object movement. How can students improve the lives of teachers?
- Session 3 Robot Pickup Service
  In this session, students invent a robot pickup service that kids can use to quickly get items from a smart locker. The locker could store food, books, or gym equipment. The process will be contactless. A user will receive a secret code to unlock a specific box. If the wrong code is entered, an error message will display. Students will use Scratch to build a model of their AI prototype. How can they improve the lives of kids?
- Session 4 Self-Driving Tour
  In this session, students become computer vision specialists. They design a self-driving tour. It must meet the needs of both business owners and tourists. Using Scratch, students will build a program that drives an autonomous vehicle along a route from one exhibit to another. At each stop, a robot will share interesting facts. The tour could take place at a zoo, theme park, or city center.
- Session 5 Obstacle Detection
  In this session, students improve the safety of their self-driving tour. They program their autonomous vehicle to avoid obstacles along the route. Afterwards, both business owners and tourists test the design to provide feedback. Once it is ready, students invite others to take a robot guided tour and provide a customer review. How does the invention enhance the lives of others?
- Session 6 Investment Opportunity
  In this session, students create a presentation for potential investors. They are seeking funding for one of their AI prototypes. It could be their drone delivery system, robot pickup service, or self-driving tour. Getting straight to the point they will explain their product. Using very few words and lots of visuals they will summarize how the technology works and why it improves the lives of users. Who will invest in their invention?

Extension Activities:

Be a Responsible Digital Citizen, Learn About Drones and AI, Upload a Sprite, The Road to Driverless Vehicles, Organize Code Area, Print Handouts

Technology Skills: artificial intelligence, programming, presentation

Technology Integration: Computer Science, Science

Software Applications: Scratch | PowerPoint or Slides

TechnoCode

In this course, students become coders that design a fun Activity Studio for kids using Scratch. Through discovery and exploration, they learn how to create a series of hands-on activities that children will enjoy playing. The young programmers apply computational thinking to build algorithms that sequence commands, events, loops, and conditions. They learn how to construct scripts to develop animated scenes, mazes, interactive stories, and games. Additional challenges extend coding skills to create artwork, compose music, produce a diorama, and more!

The technology course has 32 assignments that are divided into 6 Sessions:

- Session 1 Become a Programmer
  In this session, students are introduced to programming. They design animated scenes using Scratch. To start, they consider the importance of technology in daily life. Afterwards, students study the Scratch interface to label the parts. Once familiar with the environment, they discover how to stack blocks of code together to form a script that makes a character talk. Once they have mastered some of the basics, they explore the Scratch libraries to make a scene of two friends having fun.
- Session 2 Build an Aquarium
  In this session, students create their first project for the Activity Studio. It is an animated aquarium. To start, they explore Scratch Motion blocks to discover how they can be used to make sprites move across the stage. Next, students use forever and if then blocks to control the fish swimming. Afterwards, they learn how to use the Paint Editor to design a fish tank that has a custom backdrop. To practice coding skills, a list of challenges provides a creative spark. Upon completion, the project is prepared for viewers. Students are then given the option to share the file with the Scratch community and classmates.
- Session 3 Design a Maze
  In this session, students create their second project for the Activity Studio. It is a maze game. This activity provides an opportunity for students to practice coding skills from Session 2 to solidify their learning. To start, they complete a planning sheet to organize their ideas. Next, they use Scratch to create a puzzle that has players help a character find a way to the end of a path using arrow keys. To make the project unique, a list of challenges helps to make a one-of-a-kind maze. Upon completion, the game is prepared for players.
- Session 4 Broadcast a Story
  In this session, students create their third project for the Activity Studio. It is an animated story about a magical place. To start, they explore the Looks blocks to discover how they change the appearance of the main character and setting. Next, they enhance storytelling by triggering actions to occur when there is a switch in the backdrop. Afterwards, they direct the timing of events by sending messages to sprites using the Broadcast blocks. To practice coding skills, a list of challenges provides a creative spark.
- Session 5 Engineer a Game
  In this session, students create their final project for the Activity Studio. They apply their coding skills to develop a game. To start, they use planning sheets to determine the objective, scoring system, timing, and coding structure. Next, they discuss their design with a partner to assess if it is suitable for young children. Afterwards, students follow instructions to build and test the code. Challenges are included to foster originality. Upon completion, the project is prepared for players.
- Session 6 Curate an Activity Studio
  In this session, students build an Activity Studio for kids. It will have a collection of Scratch projects including an animated scene, maze, story, and game. To gain player feedback a link to the studio will be shared. Based on observation and questioning, students make recommendations for improving their Activity Studio.

Extension Activities:
Edit Your Scratch Public Profile, Draw Artwork with the Pen, Invent an Instrument, Record a Sound Clip, Organize Scripts with Broadcast, Chat with a Sprite, Remix a Scratch Project

Technology Skills: programming

Technology Integration: Computer Science, Mathematics, Language Arts, Science, Social Studies

Software Applications: Scratch

TechnoHTML5

In this course, students become web designers using HTML and CSS. Students are introduced to the history of the Internet and HTML language. Next, they manipulate source code to discover the function of common tags and CSS attributes. This knowledge is applied to construct a web page. Throughout the design process instructions explain how to style text, graphics, and hyperlinks to produce an informative web page. For those in need of a challenge, extension activities encourage students to create a list, apply a picture background, customize hyperlinks, employ CSS classes, insert animations, and build tables. Upon completion, the files are uploaded to the Internet.

The technology course has 18 assignments that are divided into 6 Sessions:

- Session 1 Seeking the Source
  In this session, students gain an understanding of the Internet and HTML. To begin they learn some basic terminology and consider the importance of the Internet in their lives. Afterwards, they read about key events in the development of this amazing technology. Once familiar with the history, students learn about the WWW and use a web browser to view web pages. They are introduced to HTML and its importance in the world today. Afterwards, they view the source code for web pages and decode the meaning. Upon completion of this session, students should have a basic understanding of Internet terminology and HTML.
- Session 2 Cracking the Code
  In this session, students learn about the function of HTML tags and CSS attributes. To start, they open a web page and edit coding for the title, headings, paragraph, line break, horizontal rule, image, and hyperlink. Once familiar with basic elements, they explore how to format a web page. They edit CSS to alter color, width, font, font size, line height, alignment, and padding. Once students understand the structure of an HTML document, they plan the design of their web page. They search the Internet for information, pictures, and hyperlinks. This provides them with the material they need to start building their web page in the next session.
- Session 3 Begin Web Page Building
  In this session, students begin to construct their web page. To start, they form the structure of the HTML document. They then add the title, heading, several paragraphs, and a horizontal rule. Upon completion the web page is saved as index.html. Next, students format each element using CSS. They set the color, width, position, font family, font size, text alignment, padding, and line height. Tips are offered to help students improve the appearance of their web page.
- Session 4 Add Images and Hyperlinks
  In this session, students continue to design their web page. To start, they study their saved pictures to make sure they are appropriately named and resized. Once their images are prepared, students use HTML to insert the pictures and format the style using CSS. Next, students create hyperlinks to three websites. Additional styling challenges are included in the extension activities.
- Session 5 Meta Tags
  Students are introduced to meta tags. They explore how search engines and social media services use this information. Description and keyword meta tags are inserted into their HTML document. Next, students examine their web page using a checklist to highlight areas that require improvement. If time permits, additional challenges are in the extension activities including how to insert animations and tables.
- Session 6 Upload the Web Page
  In this session, students register for a free website account with a web hosting service. Then they upload the web page that they created and the associated image files. Finally, the class has an HTML Developers Conference in which they view each other's web pages and write compliments about the sites.

Extension Activities:
Analyzing Websites, Format Lists, Add a Background Image, Format the Hyperlink Style, Create a Class, Add Animated Text and Images, Create a Table, Submit Your Site to a Search Engine

Technology Skills: Programming, Web Design

Technology Integration: Computer Science

Software Applications: Text Editor

TechnoPython

In this course, students are introduced to Python. They complete coding missions to develop the characteristics most valued in a programmer. To start, they ignite their curiosity by exploring scripts to discover how they are put together. Next, they create games including Pet Monster Rescue, Guess It, and Adventure Quest. These foster logical thinking, persistence, and creativity, and are ideal for beginners. Upon completion, students share their favorite program in a coding presentation. Have your students develop original code using loops, functions, and conditionals.

The technology course has 24 assignments that are divided into 6 Sessions:

- Session 1 Into the Coding Jungle
  In this session, students explore the Coding Jungle. The goal of this mission is to learn about Python. To start, the explorers are introduced to terminology by experimenting with code. Once familiar with the role of a programmer, they play a Python Hunt game and then edit the program to discover how it works. Afterwards, they break code in the Catch the Bugs game to develop essential debugging skills. Successful completion of the four-part mission requires curiosity, which is a highly valued trait in a programmer.
- Session 2 Pet Monster Rescue
  In this session, students create a program for the Pet Monster Rescue, which is a group that finds loving homes for monsters. To prepare for the programming mission, students learn about strings, integers, and variables. They apply this knowledge to personalize the adoption process. To pair a pet owner to their monster, the programmers write code that ask questions. The answers are used to match people to their ideal pet. This is done by combining logical operators, if and else statements, and a variable that changes from True to False. Throughout the four-part mission, an emphasis is placed upon thinking logically.
- Session 3 Guess It Game
  In this session, design a guessing game in which the player must correctly pick a number before they run out of chances. Clues tell them if their answer is too high or low. This programming mission has six parts. To prepare, students first explore how to code while and for loops. Once familiar with how to repeat a set of instructions, they start to build Guess It. To guide development, the Python programmers sequence steps into algorithms. These flowcharts provide a framework for constructing each part of the program. Fun challenges encourage students to build a unique game. Interwoven throughout all tasks is a focus upon being methodical. This skill helps programmers test different cases to solve problems within the code.
- Session 4 Adventure Quest Part 1
  In this session, students develop a text-based adventure game. It is a quest that has players overcome challenges to earn rewards. To prepare for this programming mission, students learn techniques to standardize data entry. Next, they apply these skills to build the first part of their game. It will allow players to pick a direction to explore. It will also include a challenge whereby the player can win coins when they travel North. To complete the task, students must be persistent. What will happen in this strange land?
- Session 5 Adventure Quest Part 2
  In this session, students complete their text-based adventure game. They develop a treasure hunt that has players travel East to collect objects. They must avoid danger, or risk losing it all! To prepare for this part of the programming mission, students learn about lists. They add, remove, sort, and count items. Once this skill is mastered, they apply it to their quest. Throughout the activities, an emphasis is placed upon creativity. This trait is essential as it allows programmers to design original programs.
- Session 6 Coding Presentation
  In this session, students share their favorite Python program in a coding presentation. They demonstrate how the game works and explain the code. This provides an opportunity to develop strong communication skills, which help programmers do their job.

Extension Activities:
Coding Reflections, Find and Fix the Bugs, Open a Pet Monster Picture, Keep Score, Toss a Rare Coins, Game Test the User Experience, Earn a Reward, Create a Map

Technology Skills: programming

Technology Integration: Computer Science, Mathematics, Language Arts, Social Studies

Software Applications: IDLE Python 3

Coding Collection Course Descriptions: Senior (Grades 8-12)

TechnoChatbot AI

In this course, students become chatbot developers. They use both coding and a free bot-builder to design chatbots. To start they explore the practical applications of AI technology. Once familiar with the possibilities of natural language processing, they design their own bots. First, they make a rule-based chatbot that answers event attendees' questions using keyword matching. Next, they code a transactional chatbot that takes fundraiser orders. Finally, they train a virtual agent to recruit new members.

The technology course has 20 assignments that are divided into 6 Sessions:

- Session 1 Introduction to Chatbots
  In this session, students explore the practical applications of chatbots. They chat with real business bots to critically examine strengths and limitations. Later, they prepare to create their own chatbots by registering for a Scratch account. This will be used in upcoming Sessions to build an event and order chatbot. To extend learning, an optional activity provides insight into key historical moments in chatbot development.
- Session 2 Design an Event Chatbot
  In this session, students build a chatbot for a school event using Scratch coding. It could be a prom, dance, spirit day, talent show, or retirement party. The chatbot will answer attendees' questions. To make the bot appear intelligent, it uses if-then logic and keyword matching. Upon completion, students evaluate their chatbots' limitations. An optional activity about AI expands students' knowledge of large language models.
- Session 3 Create an Order Chatbot
  In this session, students create an order chatbot using Scratch coding. It is a transactional chatbot, which can do one action. It will simplify the task of taking school fundraiser orders for items such as hats, cookies, or flowers. Just like a real salesperson, the bot will ask questions to help the customer make purchasing decisions. It will store their answers in a list. At the end of the conversation, the customer will be able to edit and then submit their order. This task expands students' knowledge of the practical applications of chatbots. In addition, it also introduces the use of lists, if-else logic, and broadcasting. An optional activity about support tickets develops a deeper understanding of how chatbots can automate tasks.
- Session 4 Design a Virtual Agent
  In this session, students use the bot-builder app, Bot Libre, to design a virtual agent for a school or an organization. It will provide help to a new student or member. The chatbot will use natural language processing to answer common questions, provide directions, connect a user to information sources, and more. To start, they select a topic such as a local drama club, sports team, or art class. Next, they create a bot and then chat with it to discover what it does and does not know. Afterwards, they select a voice that will appeal to users. An extension activity emphasizes the need to be a responsible digital citizen by examining the Bot Libre terms of use and privacy policy.
- Session 5 Train the Virtual Agent
  In this session, students train their virtual agent to increase its intelligence. To start, they view pre-programmed responses to verify the chatbot's existing knowledge base. Next, they add a new greeting and modify default responses to make the bot friendly and helpful. The training continues by adding questions and answers for common requests such as directions and contact information. Students then share their chatbot with a friend to test the design. They analyze the conversation to improve responses.
- Session 6 Deploy the Virtual Agent
  In this session, students deploy their virtual agent. To prepare they use a checklist to verify that the chatbot can complete essential tasks. Once they are certain it is ready to use, they share their bot. This allows new members or students to get the information they need about an organization.

Extension Activities: History of Chatbots, AI Chatbots and You, Insert a Saved Image as a Sprite, Export and Print the Order List, Be a Responsible Digital Citizen

Technology Skills: Artificial Intelligence, Coding, Data Analysis, Digital Citizenship, Graphics
Technology Integration: Business Studies, Computer Science
Software Applications: Scratch, Bot Libre

TechnoFuture AI

In this course, students create an interactive 'Choose Your Own Adventure' story set in a world shaped by artificial intelligence. They start by researching emerging technologies. Students transform these facts into science fiction that imagines the future. To bring this reality to life, students use AI tools to generate images, 3D models, and music. They also program a virtual tutor in Scratch that invites reader participation. Through TechnoFuture AI, students explore the applications of AI and reflect on ethical questions.

The technology course has 32 assignments that are divided into 6 Sessions:

- Session 1 Become a Futurist
  In this session, students take on the role of futurists, imagining tomorrow. They begin by exploring artificial intelligence and discovering how it is already part of their everyday lives. Then they watch a future-themed video that brings the possibilities to life. Using an AI-powered research tool, students investigate emerging innovations and record their discoveries. Finally, they team up with classmates to build a bubble map packed with creative inventions for homes, schools, and workplaces of the future.
- Session 2 Illustrate the Home of the Future
  In this session, students kick off their 'Choose Your Own Adventure' sci-fi story by inventing a futuristic gadget designed to help teens get ready for school. They begin by imagining the purpose, appearance, and user interface of the device. Then, they bring their invention to life using an AI Image Generator, crafting visual representations through descriptive text prompts. After creating their gadget, students write the opening scene of their story, which starts with the reader waking up in a high-tech home of the future. An optional extension activity has students explore ethics related to the use of AI-generated artwork.
- Session 3 Model Futuristic Transportation
  In this session, students offer readers two unique ways to travel to school. They start by designing a futuristic vehicle such as a hover scooter or teleporting bubble based upon an image found online that has a Creative Commons license. Using a 3D AI modeling tool, they transform the flat image into a 3D object. Next, they create a slide that illustrates the journey from home to school using their invention. During a gallery walk, classmates explore each other's creations and select a second futuristic vehicle to include in their story. By sharing 3D models and slide content, students give readers engaging, imaginative choices. As an optional extension, students explore AI cybersecurity systems that could protect school data.
- Session 4 Code a Virtual Tutor
  In this session, students imagine schools in the future. They investigate how computer vision and artificial intelligence could transform learning. Next, students design a school day itinerary and incorporate it into their story, featuring high-tech activities. To simulate an AI-powered testing environment, students use Scratch to create a True or False quiz. The program allows users to respond by swiping their hands over virtual buttons, mimicking gesture-based input. The quiz provides real-time feedback by automatically checking answers and displaying results. Once complete, students add a link to the quiz in their interactive itinerary so readers can take the test as part of the story. As an optional extension, students can watch a TED Talk on extended reality or create an animated 3D population map, generated with the help of AI tools.
- Session 5 Envision the Future of Fun and Work
  In this session, students finish their story. As the school day ends, the reader faces another choice: head to work or join friends. For the 'Work in the Future' path, students design a graphic that outlines a futuristic task list. For the 'Entertainment in the Future' path, students use a Generative AI tool to create an original background song that captures the scene's mood. Once both options are complete, students write an ending. As an optional extension, students can analyze a TED Talk on AI and creativity.
- Session 6 Imagine Tomorrow
  In this session, students unveil their imaginative, futuristic worlds to readers. They begin by testing the story's flow, ensuring that each choice leads to the correct slide. Next, they enhance the storytelling experience by applying advanced animation techniques. Once their interactive story is complete, students share their 'Choose Your Own Adventure'. They then read a classmate's sci-fi story and leave a comment highlighting their favorite part. To wrap up the course, students reflect on their creative journey.

Extension Activities: Risks of Artificial Intelligence; Discuss the Ethics of AI Artwork; Cybersecurity in the Future; Explore Extended Reality; Generate a 3D Map; AI, Creativity, and Voice Cloning; Improve Accessibility; Create a Printed Book

Technology Skills: Artificial Intelligence, Presentation, Coding, Word Processing, Spreadsheet
Technology Integration: Computer Science, Language Arts, Science
Software Applications: Generative AI Tools, PowerPoint, Word, Excel, Scratch

Contact Information

TechnoKids Inc. offers free curriculum support.
Contact our support staff by email at support@technokids.com or by telephone 1-800-221-7921.

We want to be your partner in computer education. If you have any comments or questions regarding our instructional materials, please contact our Head Office.

TechnoKids Inc.
55 King Street East
PO Box 451
Thornbury, ON N0H 2P0

information@technokids.com | support@technokids.com

Toll Free: 1-800-221-7921 (North America only)
Phone: 905-631-9112