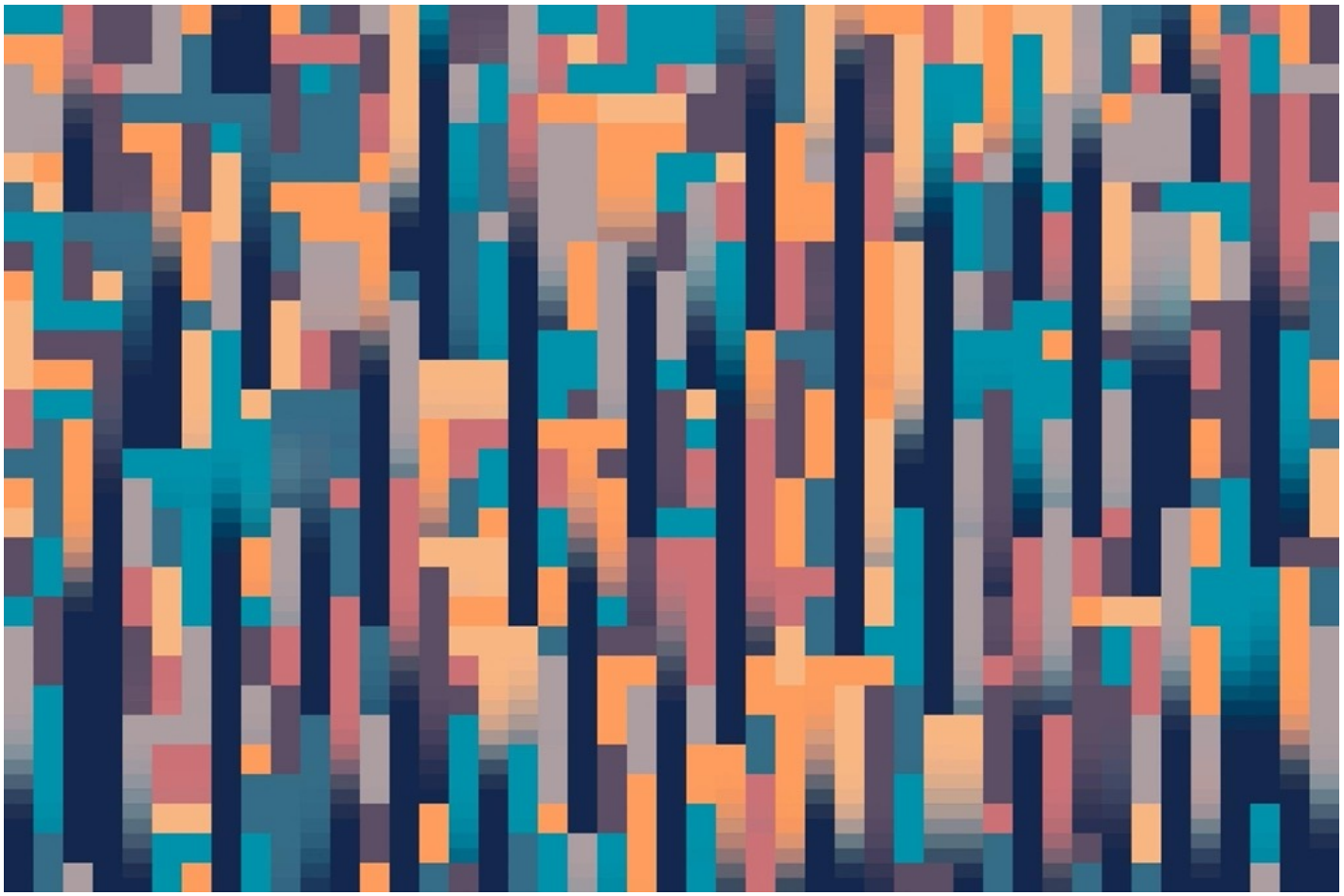
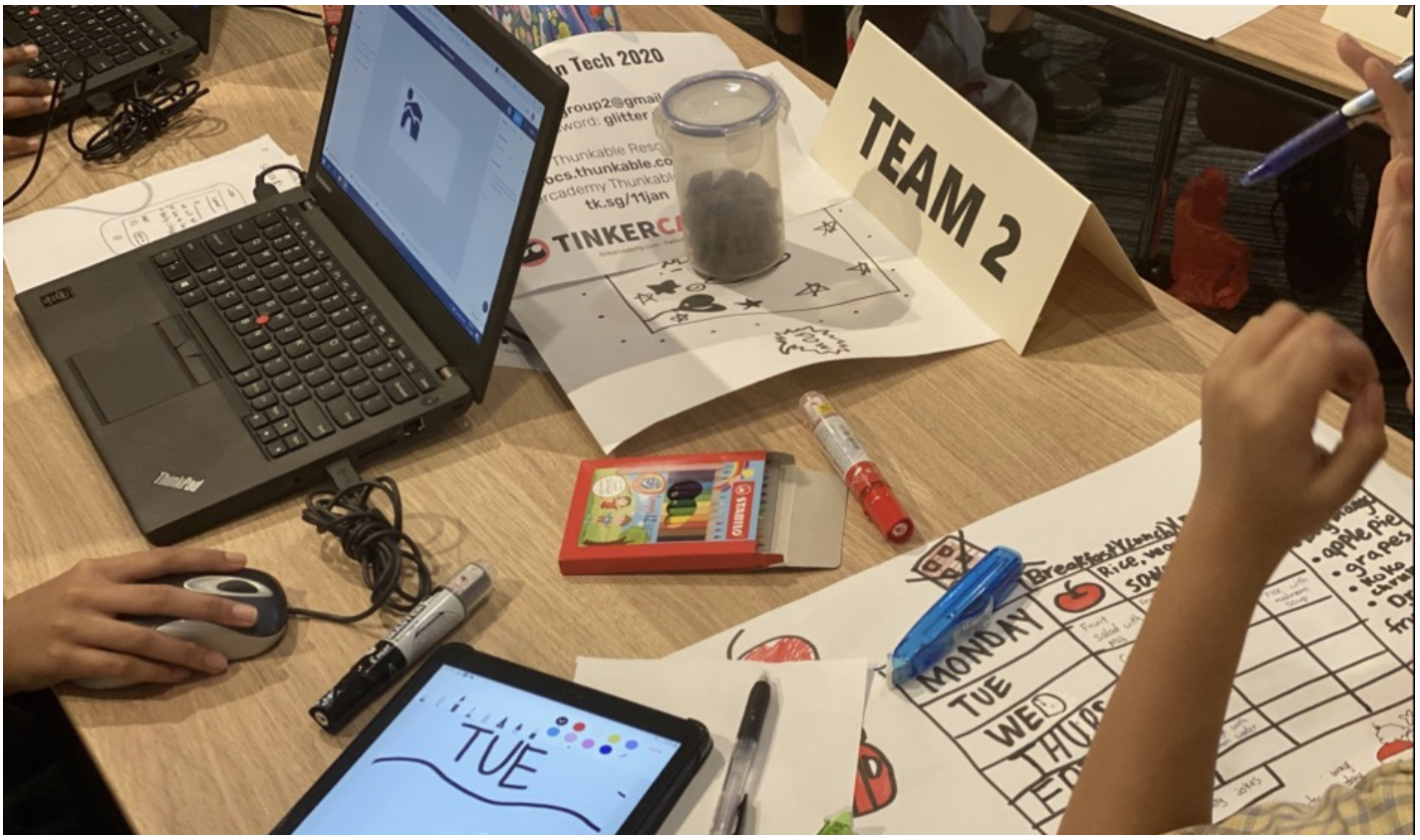
COMPUTER SCIENCE, APPLIED TECH AND STEAM EDUCATION

Programme Catalogue for Primary, Secondary and JC // 2023



TINKERCADEMY

6917 6920 • hello@tinkercademy.com • tinkercademy.com



CODING FOR ALL

We strive to inspire a love for technology through fun, hands-on, realistic, and engaging activities that empower learners to create.

We can't entirely predict what new technologies or industries will be dominant in the coming years, but we can future proof our learners by equipping them with the skills to successfully navigate the changing world.

To us, picking up technology skills is not about learning a particular programming language or platform. What we deeply care about is leading learners on a journey towards curiosity and empowerment. We consciously design our programmes to instil confidence in our learners. Through time-tested and creative teaching pedagogies, we seek to build competencies such as critical thinking, creativity, problem solving, collaboration, and learning how to learn skills. For each group of learners, we select from a wide range of content and platforms in our toolbox, so that the course is both age- and skill-appropriate for the participants. Technology is used both as a tool and medium for more holistic learning.

We wholeheartedly believe that technology education unlocks the potential for anyone to become meaningful creators and changemakers. So partner with us to help empower the future thinkers, innovators, and do-ers of our world.

TINKERCADEMY

OUR COURSES

Mobile App Development

#	Courses	Difficulty	Audience
1	Mobile App Development with Thunkable	✦✦	All Ages
2	Mobile App Development with SwiftUI	✦✦✦	Sec/JC
3	Mobile App Development with React Native	✦✦✦	Sec/JC

Future Technologies

#	Courses	Difficulty	Audience
1	Unleashing the Potential of Blockchain Technology	✦	All Ages
2	Building the Metaverse with Augmented and Virtual Reality Experiences	✦✦	All Ages
3	Connecting to the Internet of Things (IoT) with M5Go	✦✦	Sec/JC
4	Cryptography for Cybersecurity	✦✦✦	Sec/JC

Data Science, AI and Machine Learning

#	Courses	Difficulty	Audience
1	Machine Learning with Teachable Machine	✦	All Ages
2	Machine Learning with Scratch	✦✦	All Ages
3	Machine Learning with TensorFlow	✦✦✦	Sec/JC
4	Data Science and Analytics with Python	✦✦✦✦	Sec/JC

Digital Making

#	Courses	Difficulty	Audience
1	Digital Making with littleBits	✦	All Ages
2	Digital Making with Micro:bit (MakeCode)	✦✦	All Ages
3	Digital Making with Micro:bit (Python)	✦✦✦	Sec/JC
4	Digital Making with Arduino	✦✦✦	Sec/JC

Applied Learning

#	Courses	Difficulty	Audience
1	Food Science and Technology	✦✦	Sec/JC
2	Smart Cities	✦✦	Sec/JC
3	Sustainability	✦✦	Sec/JC
4	Healthcare Technology	✦✦	Sec/JC

Computer Science

#	Courses	Difficulty	Audience
1	Intro to Computer Science with Minecraft	✦	All Ages
2	Computer Science Foundations with Sphero	✦✦	All Ages
3	Generative Art and Animation with Python	✦✦	Sec/JC
4	Intro to Computer Science with Python	✦✦✦	Sec/JC
5	Python for Business	✦✦✦✦	Sec/JC

Game Development

#	Courses	Difficulty	Audience
1	Game Development in MakeCode Arcade	✦✦	All Ages
2	Game Development in Unity Bolt	✦✦✦✦	Sec/JC

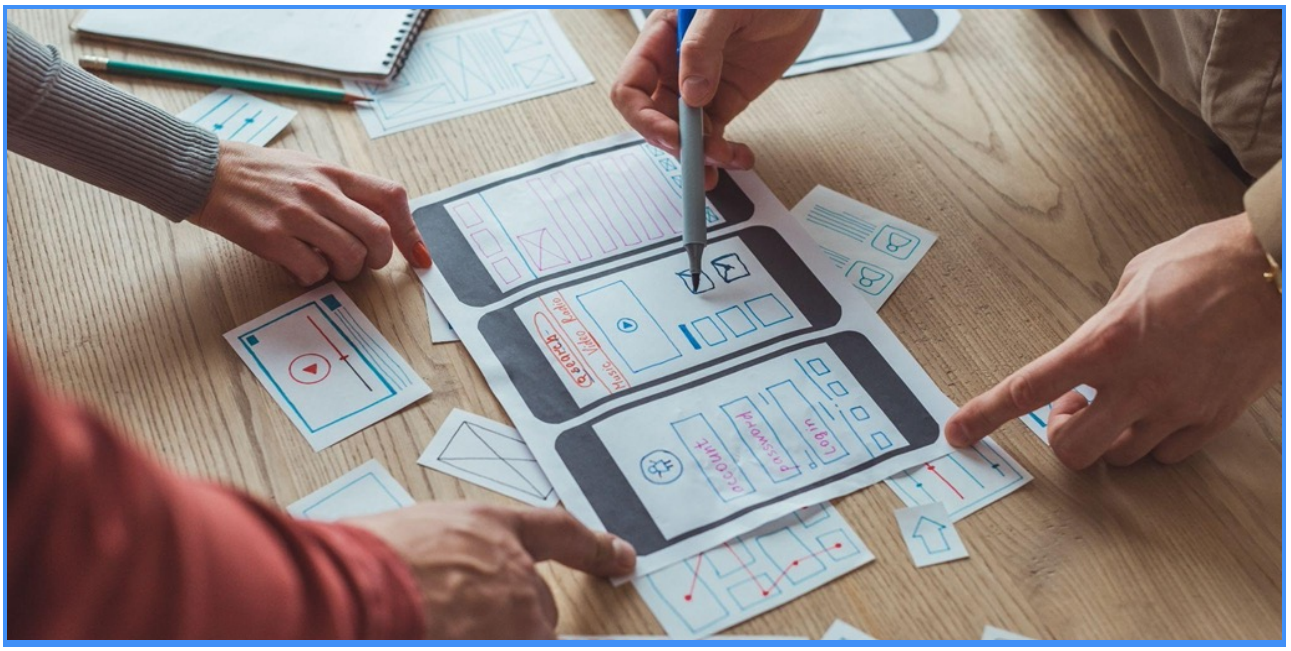
Web Development

#	Courses	Difficulty	Audience
1	Introduction to Web Technologies (HTML/CSS/JS)	✦✦	All Ages
2	Front-end Web Development with React	✦✦✦✦	Sec/JC

Engineering

#	Courses	Difficulty	Audience
1	Design, Innovation and Creative Engineering	✦✦	Primary

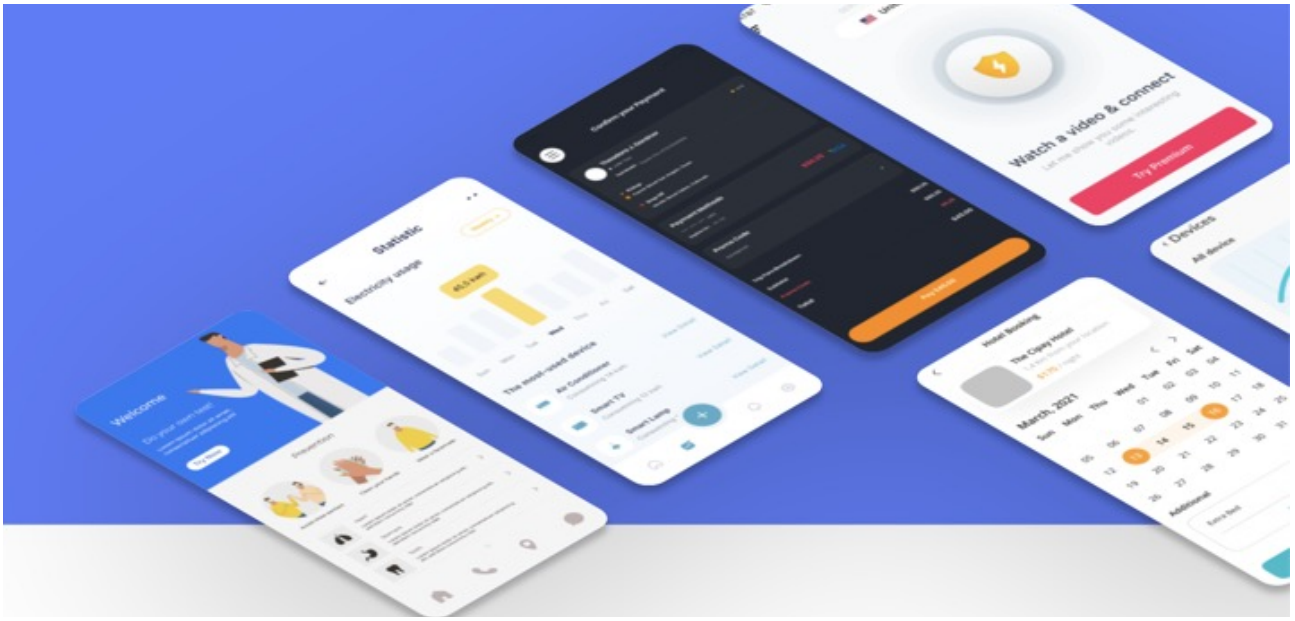
Mobile App Development



#	Courses	Difficulty	Audience
1	Mobile App Development with Thunkable	✦✦	All Ages
2	Mobile App Development with SwiftUI	✦✦✦	Sec/JC
3	Mobile App Development with React Native	✦✦✦	Sec/JC

Mobile App Development consists of a lot more than just the technology and the platform. A huge aspect of the mobile app development process is figuring out how to design user interfaces and user experiences that are simple to understand, and efficient to navigate. Oftentimes, students design the apps while only thinking of themselves as the user. However, to create apps that spark joy, students need to build empathy for their users, be willing to challenge their own assumptions, and be attentive to the small details. In our courses, we make it a point to highlight the development of these soft skills. For instance, we get students to step away from the computer screen and development environment, and conduct activities sketching multiple iterations of no-tech pen and paper prototypes, gathering feedback, and listening to their peers.

1. Mobile App Development with Thunkable



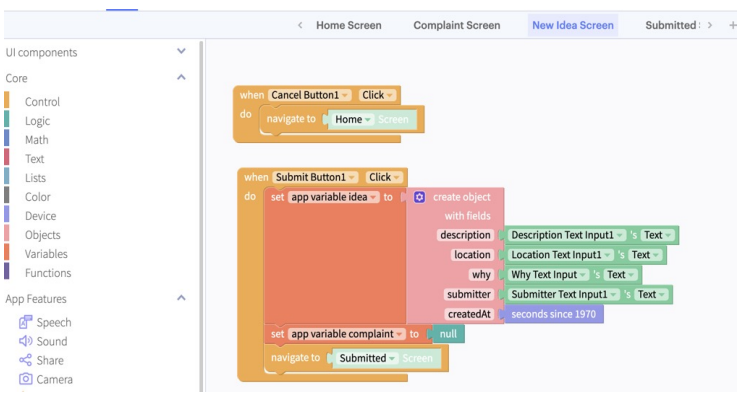
Level:	Beginner
Target group:	Secondary
Duration:	12–24 hrs
Requirements:	–
Prerequisites:	–

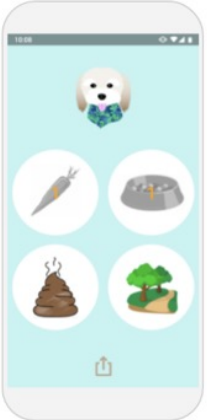
Lesson Outcomes:

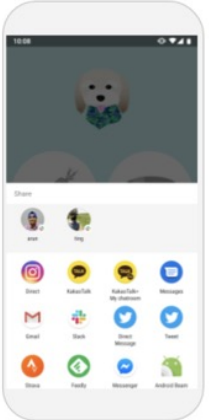
- ★ Gain critical thinking & problem solving skills
- ★ Develop UI/UX skills
- ★ Develop foundational programming skills

In this course, participants will learn to create mobile apps using Thunkable, a freely available web-based mobile development environment. The platform features a simple drag-and-drop design canvas and powerful logic blocks that allow you to unlock all the benefits of the smartphone.

Individuals will first be introduced to the basics of app development and will then explore how to develop a series of different mobile apps on Thunkable. They will also learn how to design user interfaces / user experiences. Despite the simple development environment accessible to the youngest learners, apps developed are deployable natively on both iOS and Android smartphones.







2. Mobile App Development with SwiftUI

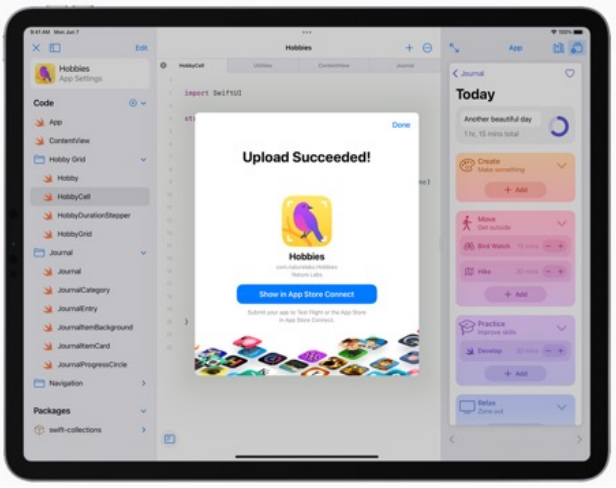
Level:	Intermediate
Target group:	Secondary
Duration:	24–36 hrs
Requirements:	iPad/Mac
Prerequisites:	–

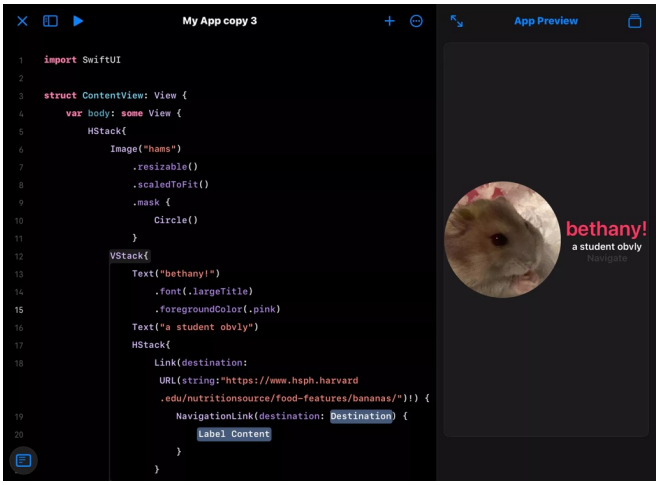
Lesson Outcomes:

- ★ Learn coding on Swift Playgrounds
- ★ Introduction to App Development
- ★ Discover the mechanisms of Apple products

As a member of the Apple Consultants Network, we are one of the leading providers of Swift programming classes for students in Singapore. In this course, individuals will be introduced to programming in the latest version of the Swift programming language, and the SwiftUI app layout framework. Swift and SwiftUI, created by Apple, are designed to be easy for beginners to learn, and powerful for advanced users to utilise in creating iOS apps.

Individuals will code in Swift Playgrounds, a freely available app on Mac and iPads, and learn about syntax-based programming concepts, while applying them to build actual projects in SwiftUI, using on-screen components in modern Apple apps. At the end of the course, participants will walk away with mini-apps that they can further develop.





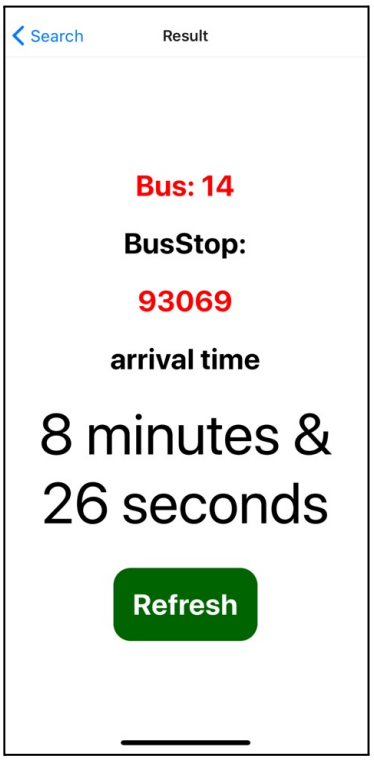
3. Mobile App Development with React Native

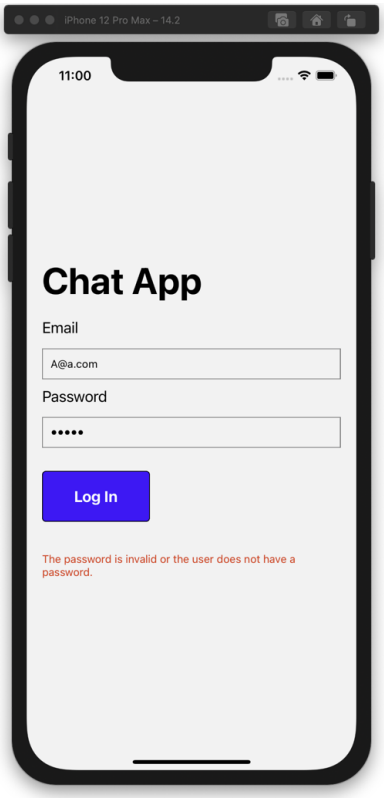
Level:	Intermediate
Target group:	Secondary
Duration:	24–36 hrs
Requirements:	–
Prerequisites:	Experience with HTML/CSS, & JavaScript if available

Lesson Outcomes:

- ★ Introduction to React Native
- ★ Further JavaScript
- ★ Learn App Development

In this course, participants will learn to build mobile applications that run across both iOS and Android devices, using React Native, a free open-source front-end framework created and maintained by Meta. Background knowledge in HTML and CSS, preferably JavaScript, will be beneficial for this course. Students will learn about various React Native features and functions to develop a series of small, useful mobile apps.





,

Swift Accelerator

Level:	Advanced
Target group:	Secondary
Duration:	50+ hrs
Requirements:	Mac required
Prerequisites:	Prior knowledge in any programming language

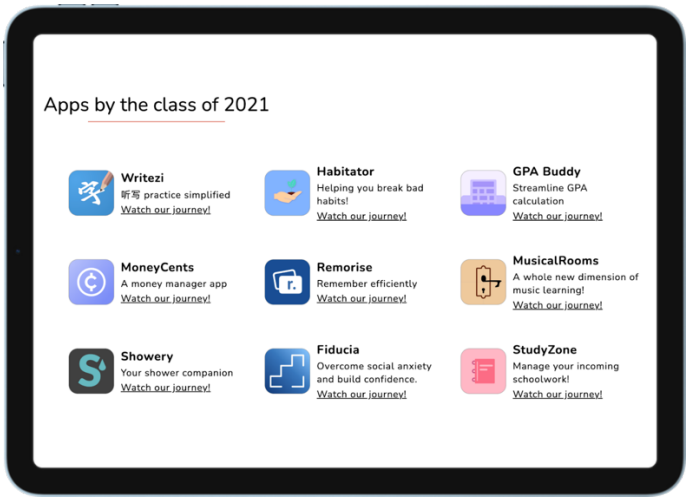
Lesson Outcomes:

- ★ Learn App Development with Swift for iOS devices
- ★ Build and release custom apps
- ★ Learn, hone, and apply design & storytelling skills

The Swift Accelerator Programme is an intensive talent development programme for secondary school students with an interest and aptitude for app development and computational thinking. This programme runs over several months and participants of the programme will learn to code in Swift, and apply story-telling and design thinking principles to design, build and release an iOS app of their own.

First launched in 2018 by IMDA and Apple, the programme has helped develop over 250 young coders by giving them the space, opportunity and guidance in coding, story-telling and design thinking, in order to help them take their passion to the next level. This programme has an annual intake in March. For more information on signing up, please visit swiftinsg.org.

"The Swift Accelerator Programme has really broadened my horizons and I really learnt a lot about app development, and not just Swift, but also UI design and other soft skills such as teamwork and how to work with people remotely :) highly recommend!" — Faith, Class of 2021



Mobile App Development Hackathon with React Native (Virtual)

Besides mobile app development workshops, we also conduct hackathons. Recently, we worked with a Singapore statutory board to organise a virtual hackathon. This hackathon was designed to encourage programming and foster interest in Science, Technology, Engineering and Mathematics (STEM) among tertiary students.

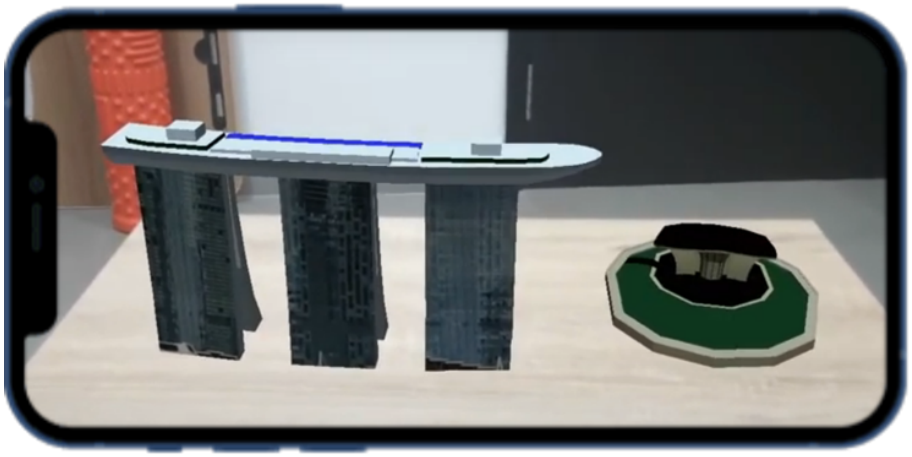
The event included a training workshop in which students learned how to create mobile applications using the React Native platform and AWS Web Services. Following the workshop, teams of four to five participants spent the next few days developing a mobile app solution that would address the problem statement.

To facilitate the learning process and the development of the application, each team was assigned a technical mentor who guided them through the ideation process and any technical difficulties they encountered.

Some feedback by the participants:

"Notes videos and lectures were detailed and uploading to Youtube allowed me to access it easily and use it as reference"

"The regular meetings with mentors should continue for future hackathons"



Future Technologies

#	Courses	Difficulty	Audience
1	Unleashing the Potential of Blockchain Technology	✦	All Ages
2	Building the Metaverse with Augmented and Virtual Reality Experiences	✦✦	All Ages
3	Connecting to the Internet of Things (IoT) with M5Go	✦✦	Sec/JC
4	Cryptography for Cybersecurity	✦✦✦	Sec/JC

Technology changes fast, and it's changing faster year by year. What was a dream just yesterday is reality today. We have developed a series of hands-on experiential courses to engage learners with emerging technologies, from blockchain to the metaverse and the Internet of Things. Cutting-edge and immature technologies have a well-deserved reputation for changing quickly and being difficult to set up and use.

Our courses are therefore designed to enable learners to create using these technologies with as little prerequisite technical knowledge as possible. With hands-on activities and a more concrete understanding of these technologies, learners will be empowered to think critically on how the technologies might evolve and impact society in the near future.

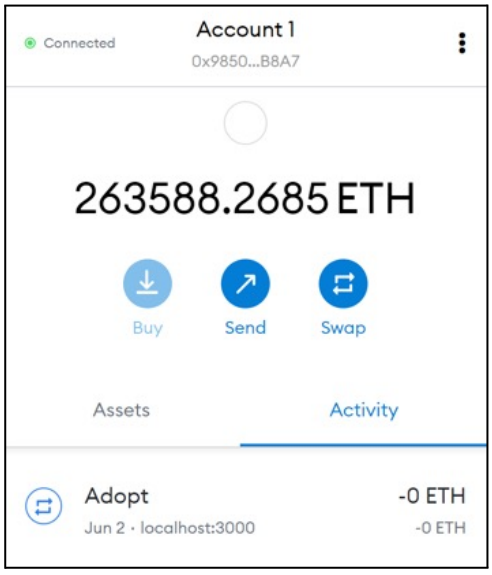
1. Unleashing the Potential of Blockchain Technology

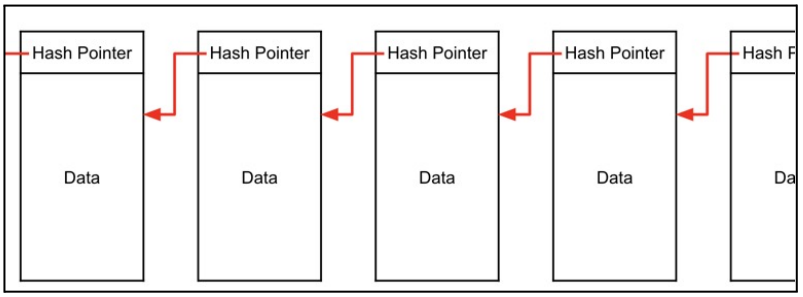
Level:	Beginner
Target group:	Upper primary & Up
Duration:	2–12 hrs
Requirements:	–
Prerequisites:	–

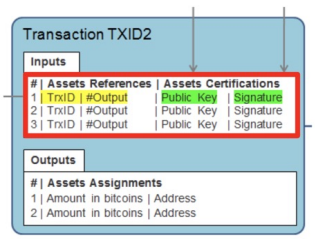
Lesson Outcomes:

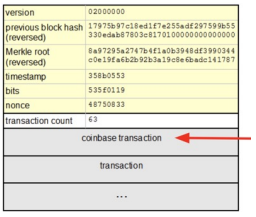
- ★ Understand Blockchain concepts through hands-on exercises
- ★ Discover the current and potential applications of Blockchain technology

Learners will explore the foundational concepts that underpin the building blocks of blockchain technology. Through multiple hands-on activities, they will unearth and highlight the concepts and technologies that make blockchain technology possible. They will also be guided to compare and contrast multiple different implementations of blockchain - including Bitcoin, Ethereum and Stellar - with the goal of demystifying the concept of blockchain technology and providing a relevant, applicable understanding. This course is particularly relevant to working professionals in domains such as finance, education and logistics. But the course can also be simplified to be suitable for a younger audience.









2. Building the Metaverse with Augmented and Virtual Reality Experiences

Level:	Beginner
Target group:	Upper Primary & Up
Duration:	2–4 hrs
Requirements:	iPads / Macs
Prerequisites:	–

Lesson Outcomes:

- ★ Distinguish between Augmented Reality (AR), Virtual Reality (VR), XR, MR, and the Metaverse
- ★ Hands-on experience building new immersive experiences in AR and VR

Learners will understand the Metaverse not as a monolithic platform, but rather as a spectrum of experiences that ranges from AR to immersive VR. They are introduced to a variety of AR and VR environments and development platforms, and will have the opportunity to better explore the possibilities in the world of AR and VR through hands-on experience with a variety of kits, applications, and authoring tools. In particular, they will design their own AR experiences on iPads or Macs with Apple's Reality Composer app.

3. Connecting to the Internet of Things (IoT) with M5Go

Level:	Beginner
Target group:	Upper Primary & Up
Duration:	12 hrs
Requirements:	M5Go IoT Kit
Prerequisites:	–

Lesson Outcomes:

- ★ Understanding what is IoT and its impact across different domains
- ★ Hands-on experience creating simplified end-to-end IoT prototypes

Learners will critically examine the potential impact of having smart devices that are connected to the Internet, whether in their daily life or in industries. They appreciate the considerations of building an IoT ecosystem, including the hardware, durability, power and network requirements for a successful IoT deployment.

As this is a hands-on course, they will also build simple IoT prototypes using the M5Go platform, such as a smart plant monitoring and watering system. The M5Go platform provides a block-based development environment that makes it intuitive for learners to program the behaviour of the unit, and it has native support for connecting to the Internet. Learners will learn to either send data to ThingSpeak, an open IoT analytics platform, or to send and receive data from an internally-developed Python-based server using a simple API.

4. Cryptography for Cybersecurity

Level:	Intermediate
Target group:	Secondary & Up
Duration:	24–36 hrs
Requirements:	–
Prerequisites:	Some programming experience, preferably Python

Lesson Outcomes:

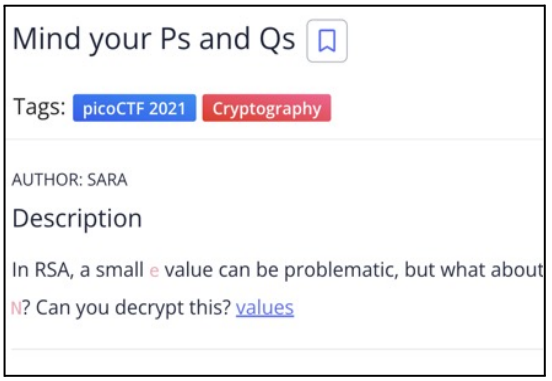
- ★ Learn about symmetric and public key Cryptography
- ★ Hands-on experience creating encryption algorithms in Python
- ★ Work with tools like Nmap, Ncat and solve CTF-style Cryptographic challenges

Learners will dive deep into the workings of a computer, to understand how a computer and networks can be compromised, and how it can be secured. This includes understanding how data is stored on a computer, how it is represented, and how it is manipulated. The course has a particular focus on cryptography, an important aspect of cybersecurity, and a rich area for practising algorithmic computational skills.

Learners will explore cryptography, in particular, early forms of cryptography that lend themselves well to hands-on activities wherein participants will code classical encryption algorithms in Python such as Caesar, Vigenere and Playfair ciphers. The course moves on to modern cryptographic protocols such as TripleDES and RSA, including bit-level understanding of how the algorithm works through hands-on programming exercises. Learners will apply their skills in a Capture-the-Flag competition against their peers, where they have to rely on their creativity and persistence to solve the challenges.

```
# xor_string will xor 2 equal length
# strings together
def xor_string(a, b):
    return [ a ^ b for a, b in zip(a,b)]

printable = [32] + list(range(65,91)) + list(range(97,122))

def is_printable(s):
    metric = 0
    for char in s:
        if not (char in printable):
            metric += 1
    return 1 - metric / len(s)

f = open('4.txt', 'r')

for line in f:
    cipher = bytes.fromhex(line.rstrip())

    for i in range(256):
        answer = []
        for b in cipher:
```

```
FOUND IT
The key is 53
7b5a4215415d544115415d5015455447414c

b'Now that the party is jumping\n'
>
```



Mind your Ps and Qs

Tags: picoCTF 2021 Cryptography

AUTHOR: SARA

Description

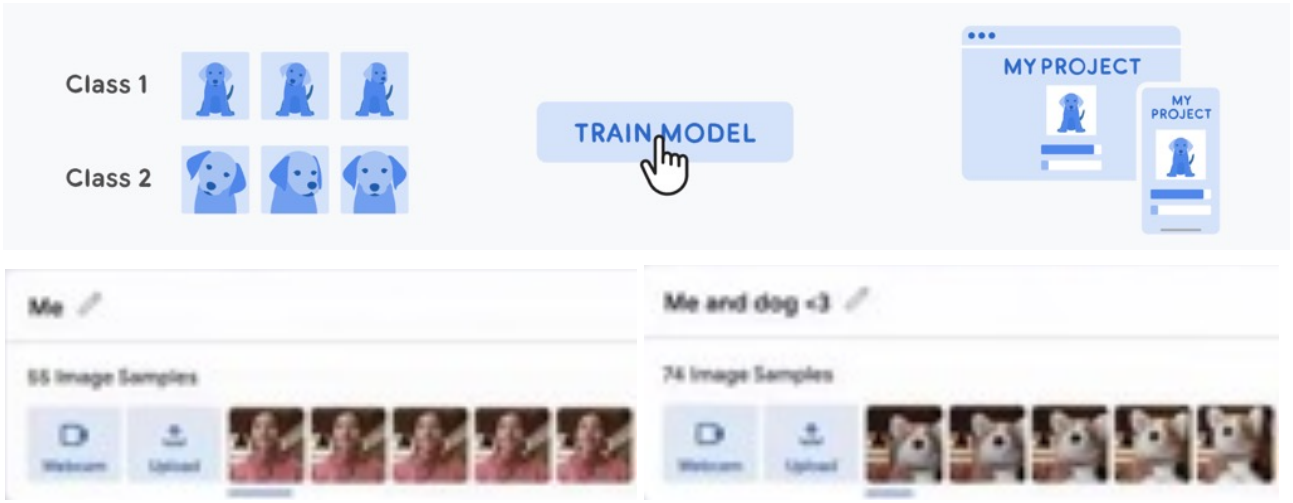
In RSA, a small e value can be problematic, but what about N? Can you decrypt this? values

Data Science, AI and Machine Learning

#	Courses	Difficulty	Audience
1	Machine Learning with Teachable Machine	✦	All Ages
2	Machine Learning with Scratch	✦✦	All Ages
3	Machine Learning with TensorFlow	✦✦✦	Sec/JC
4	Data Science and Analytics with Python	✦✦✦✦	Sec/JC

Artificial Intelligence and Machine Learning have deeply impacted our lives, and will continue to disrupt them for years to come. At the same time, the domain has never been more accessible. New platforms make it approachable not just for researchers whose primary specialty is not computer science, but also fresh learners eager to expand their horizons. Both Teachable Machine and Scratch are no-code and low-code environments to engage learners with the machine learning workflow while helping them to visualise the potential applications of AI. More experienced individuals can engage with Machine Learning and the related field of Data Science using the TensorFlow framework and the pandas and numpy libraries in Python.

1. Machine Learning with Teachable Machine

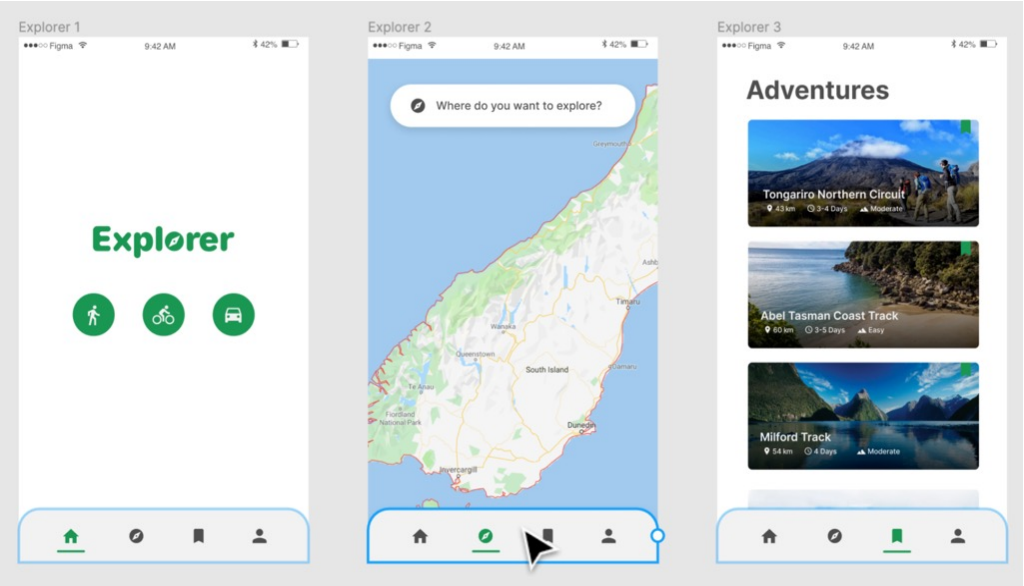


Level:	Beginner
Target group:	Secondary
Duration:	2–8 hrs
Requirements:	–
Prerequisites:	–

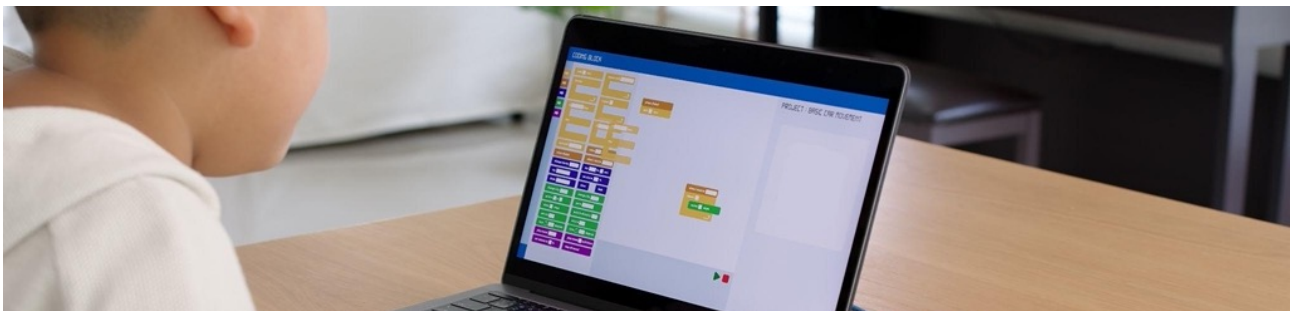
Lesson Outcomes:

- ★ Gain an appreciation of Artificial Intelligence and Machine Learning
- ★ Hands-on experience building ML applications
- ★ Learn soft skills of teamwork & communication

In this course, participants will learn the basic concepts of Machine Learning, and will work on three different projects (Image project, Audio project and Pose project) using Teachable Machine, a free web-based tool, to reinforce their understanding of the concepts. The Image project involves using images to train a model to be able to classify particular objects. The Audio project involves using short sound samples to teach a model to classify audios. Lastly, the Pose project teaches a model to classify body positions (ie. squatting, standing, stretching) using images found online or striking poses in their webcam itself.



2. Machine Learning with Scratch

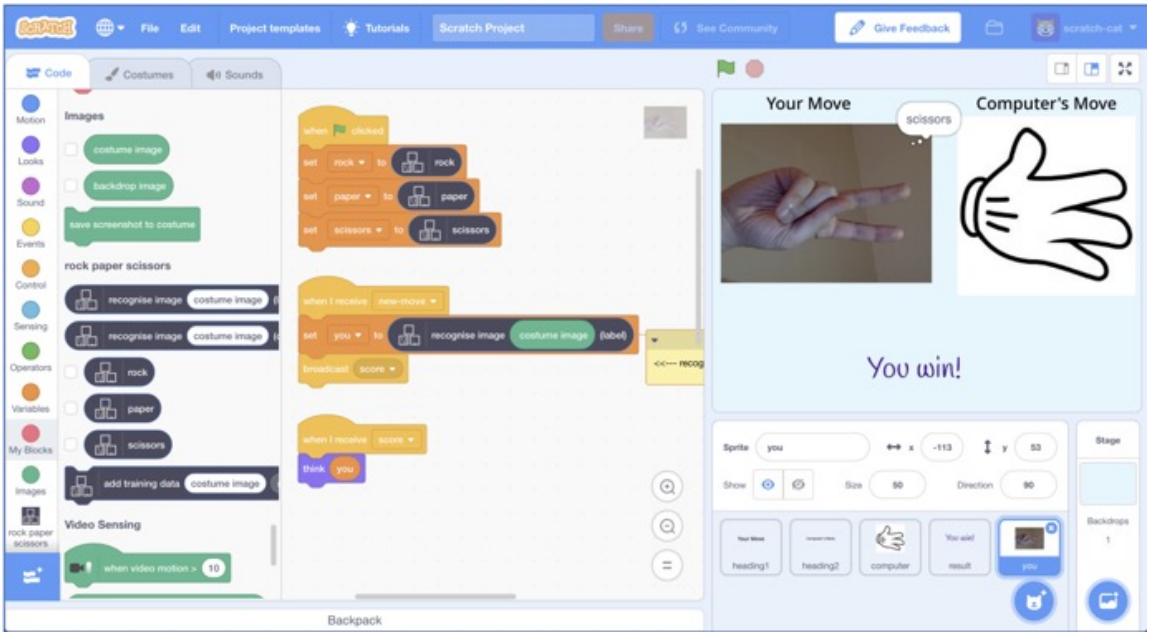


Level:	Beginner
Target group:	Upper Primary & Up
Duration:	4 hrs
Requirements:	–
Prerequisites:	–

Lesson Outcomes:

- ★ Gain an appreciation of AI & ML
- ★ Hands-on experience with Scratch
- ★ Exposure to ML concepts such as image classification and NLP

In this course, participants will learn the basic concepts of Artificial Intelligence and Machine Learning, and how to build machine learning algorithms with Scratch, a free visual programming language. The course begins with a discussion of Machine Learning concepts (such as neural networks, image classification, and natural language processing). Next, participants will delve into some hands-on activities to reinforce their understanding of these concepts. For example, individuals will be guided to build some machine learning algorithms (e.g. image classifiers), and will be tasked to work in small groups to design and implement projects of their choice.



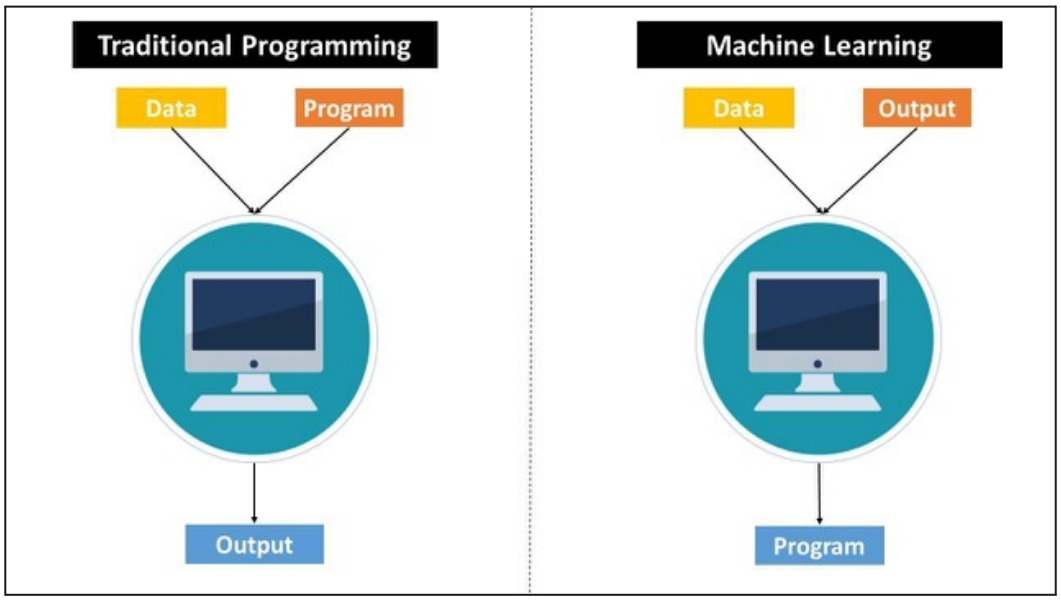
3. Machine Learning with TensorFlow

Level:	Intermediate
Target group:	Secondary
Duration:	24–36 hrs
Requirements:	–
Prerequisites:	–

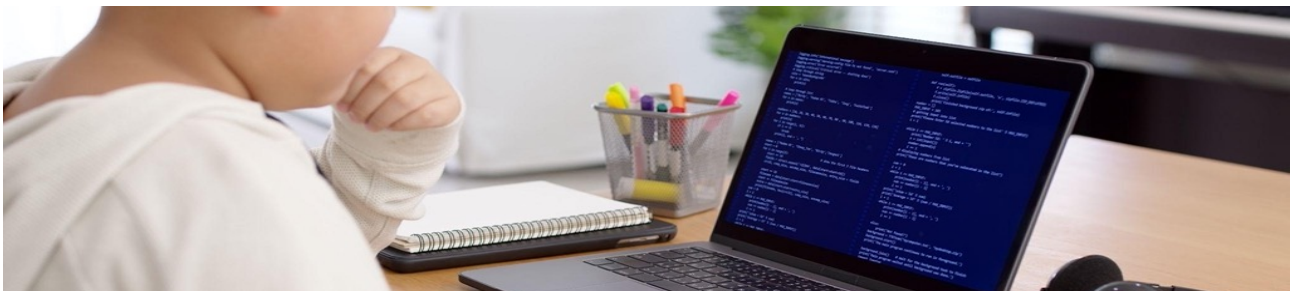
Lesson Outcomes:

- ★ Gain an appreciation of AI & ML concepts (e.g. neural networks, reinforcement learning)
- ★ Hands-on Python programming on TensorFlow
- ★ Exposure to training algorithms for image and audio classification

In this course, participants will learn how to implement Machine Learning concepts and applications (such as computer vision, natural language processing and neural networks) using TensorFlow (a syntax-based programming tool and an end-to-end open source platform for Machine Learning). Through the use of TensorFlow, they will gain a greater grasp of Python programming. This course will help to improve their overall coding abilities as they will become more familiar with syntax-based programming.



4. Data Science and Analytics with Python

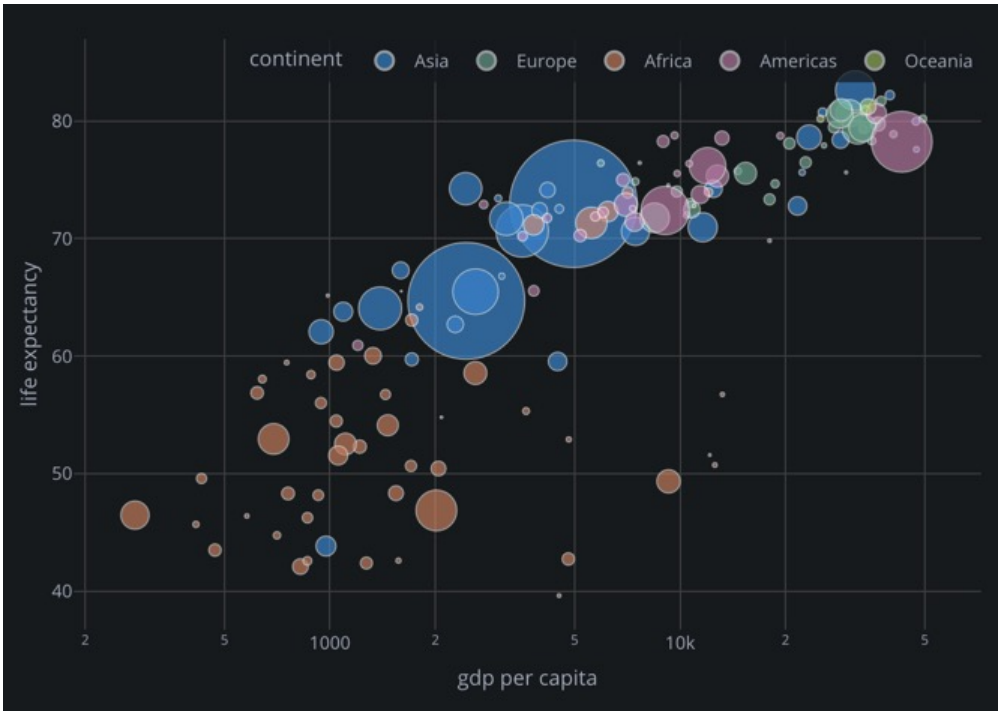


Level:	Advanced
Target group:	Secondary & Up
Duration:	24 hrs
Requirements:	–
Prerequisites:	–

Lesson Outcomes:

- ★ Introduction to analytics with hands-on Python programming exercises
- ★ Exposure to the popular pandas & matplotlib/plotly libraries

In this course, participants will learn how to use Python to explore the field of data science. Participants will be introduced to the basics of machine learning - a phenomenon that has permeated our daily lives in small and big ways. They will also learn how to build and train machine learning algorithms (e.g. image classifiers, text classifiers) using Python. In addition, individuals will be introduced to introductory data analytics using the popular pandas and matplotlib/plotly libraries in Jupyter notebooks. At the end of the course, participants will gain an appreciation of the value of data analytics, as well as the potential and pitfalls of data science.

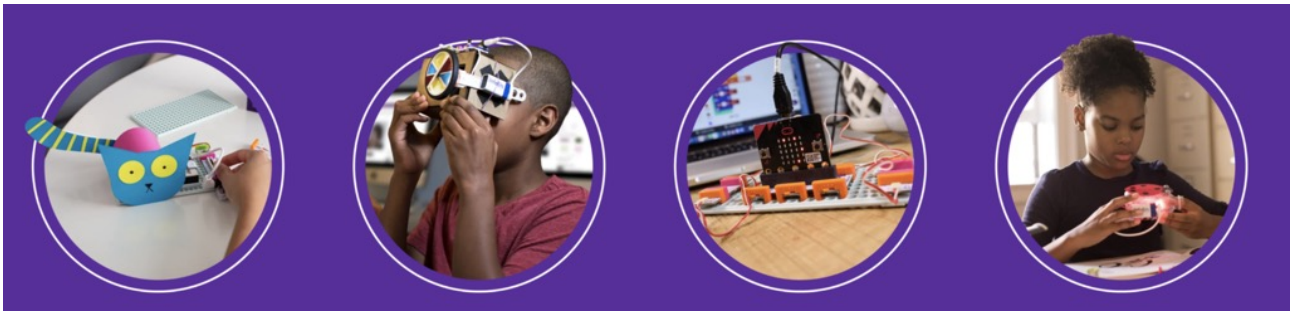


Digital Making

#	Courses	Difficulty	Audience
1	Digital Making with littleBits	✦	All Ages
2	Digital Making with Micro:bit (MakeCode)	✦✦	All Ages
3	Digital Making with Micro:bit (Python)	✦✦✦	Sec/JC
4	Digital Making with Arduino	✦✦✦	Sec/JC

Over the past decade or so, the toolchains for developing microcontrollers have become more and more accessible. Arduino was developed to be accessible for university students making interactive prototypes. Micro:bit was designed to be brought into primary and secondary school classrooms. Microcontrollers offer physical computing, with projects that are designed to interact with the real-world via sensors and actuators. They also offer a low-floor, high-ceiling approach, suitable for beginners, but powerful enough to meet the needs of experienced makers.

1. Digital Making with littleBits



Level:	Beginner
Target group:	K2 and up
Duration:	2–4 hrs
Requirements:	littleBits Kit
Prerequisites:	–

Lesson Outcomes:

- ★ Gain an appreciation of electronics and engineering
- ★ Hands-on programming and making

In this course, participants will learn how to program using littleBits. The littleBits kits consist of an assortment of various bits which snap together with small magnets for prototyping and learning. Over the duration of the course, participants will learn about the different components of an electronic device. They will then be challenged to use what they have learnt to build small electronics projects such as an automatic night light or a metronome. Such hands-on activities will be great for introducing individuals to electronics and engineering.

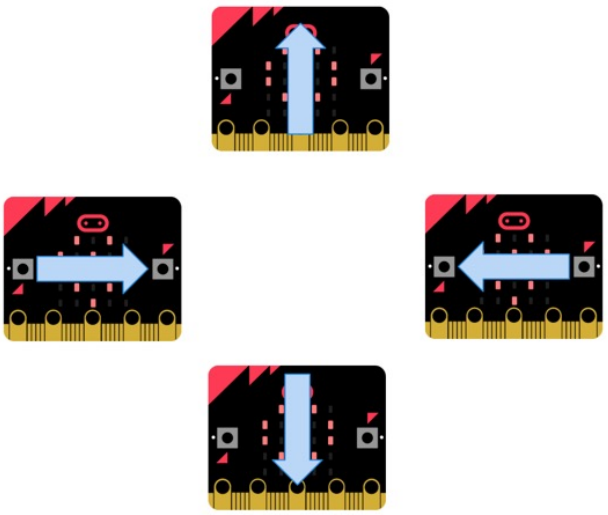
2. Digital Making with Micro:bit (MakeCode)

Level:	Beginner
Target group:	Upper Primary & Up
Duration:	4–24 hrs
Requirements:	Micro:bit Kit
Prerequisites:	–

Lesson Outcomes:

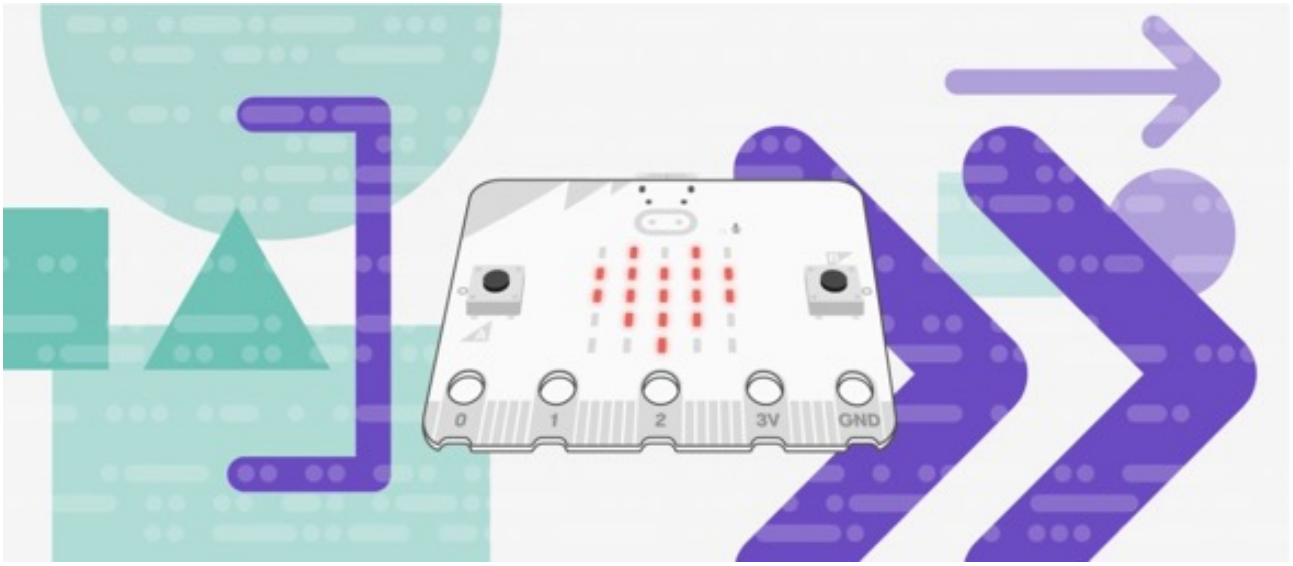
- ★ Appreciate electronics' role in our lives
- ★ Program a microcontroller to perform tasks, control outputs, and sense inputs
- ★ Ideate and build working physical prototypes with the microcontroller, components, and other materials

In this course, participants will learn how to program using code blocks in Microsoft MakeCode, a free, open source programming platform. Similar to Scratch, Microsoft MakeCode provides graphical, beginner-friendly, drag-and-drop code blocks. This course will be a good way for participants to practice thinking like a computer scientist without actually having to use syntax-based programming. Over the duration of the course, individuals will create five projects (scissors-paper-stone game, step counting device, digital compass, music machine, timing gates) on the micro:bit platform using MakeCode.



"Tinkercademy has always been at the forefront of training students and educators in microcontrollers. Their expertise in the hardware aspect had also led them to design hardware solutions to extend limited functionalities of some microcontrollers such as the micro:bit."
– Aurelius Yeo, HOD Ed Tech,
School of Science and Technology

3. Digital Making with Micro:bit (MicroPython)

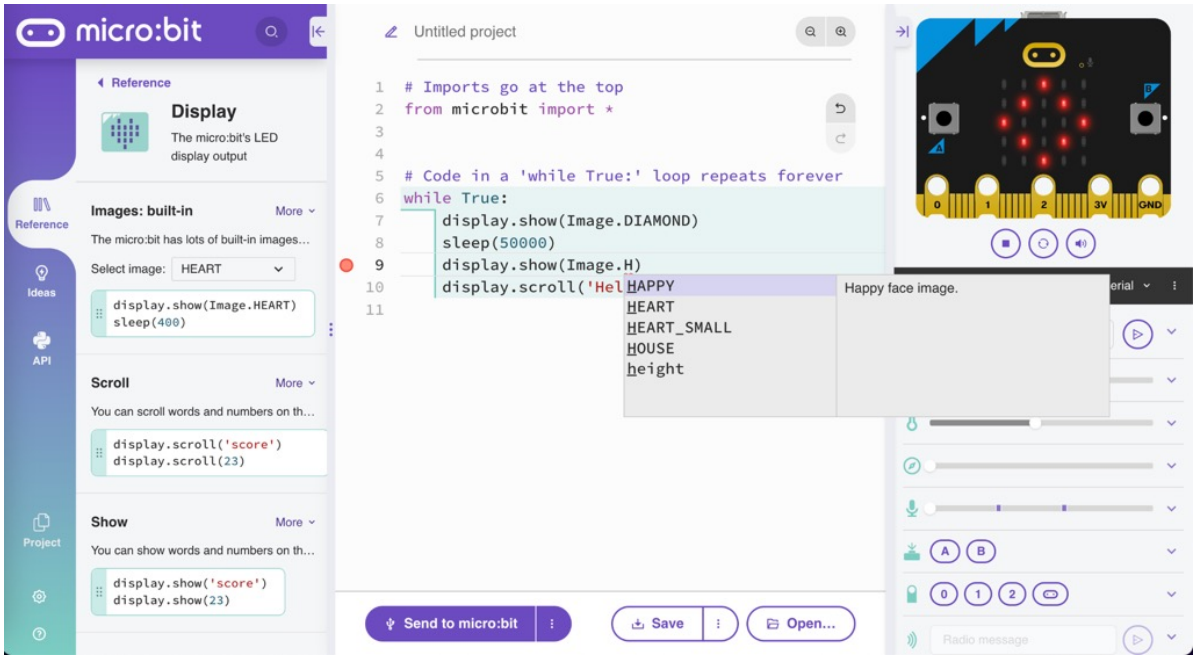


Level:	Intermediate
Target group:	Secondary & Up
Duration:	24 hrs
Requirements:	Micro:bit Kit
Prerequisites:	–

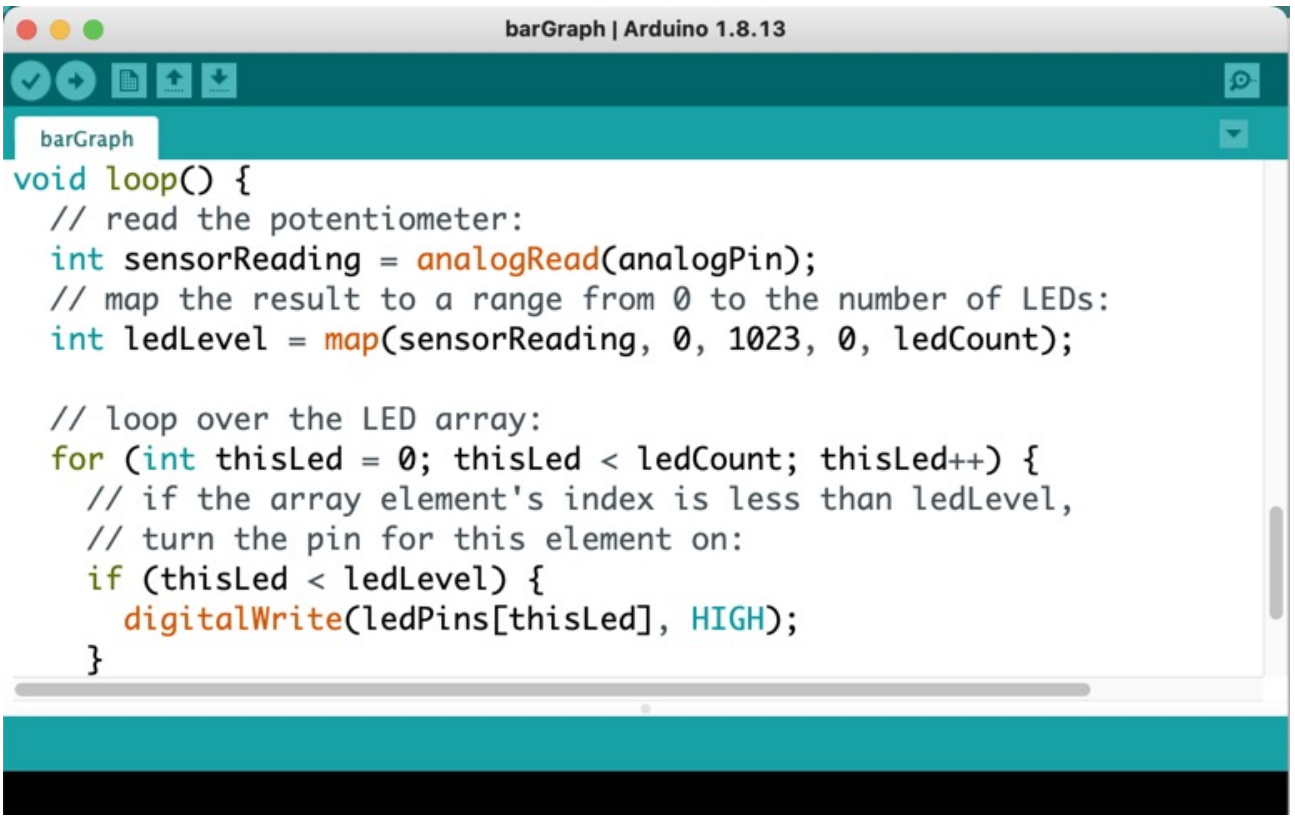
Lesson Outcomes:

- ★ Hands-on Python programming experience with the Micro:bit
- ★ Program a microcontroller to perform various tasks, control outputs, and sense various inputs

In this course, participants will learn how to program using syntax-based programming in MicroPython, a free open-source software that runs on microcontrollers. Over the duration of the course, participants will create small-scale projects on the micro:bit. This course will help to improve their overall coding abilities as they will become more familiar with syntax-based programming.



4. Digital Making with Arduino

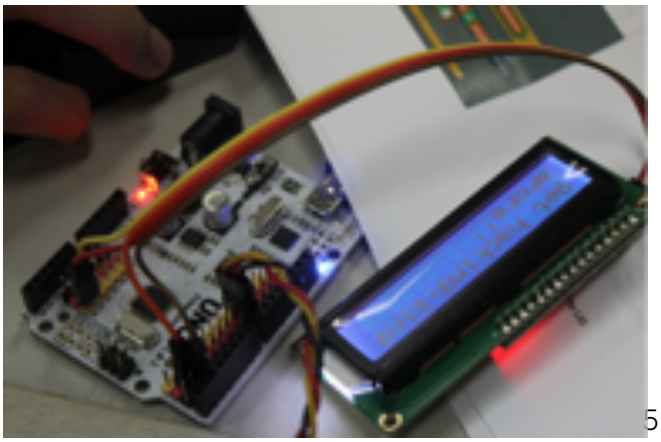


Level:	Intermediate
Target group:	Upper Secondary & Up
Duration:	24 hrs
Requirements:	Arduino Kit
Prerequisites:	–

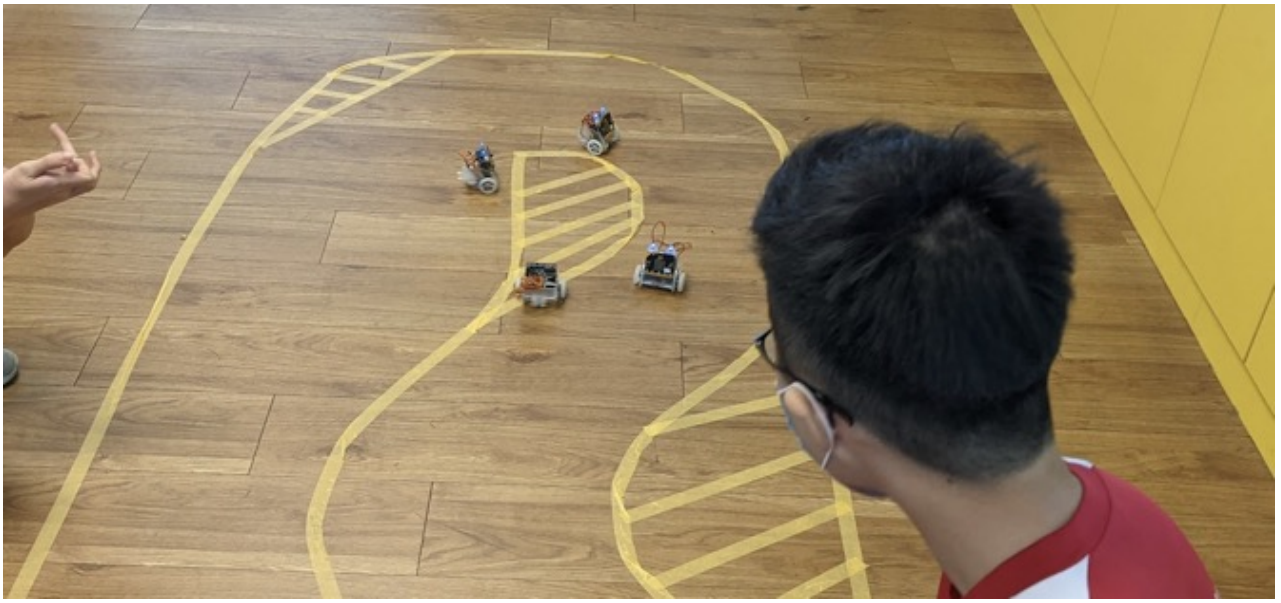
Lesson Outcomes:

- ★ Hands-on programming experience with the Arduino
- ★ Program a microcontroller to perform various tasks, control outputs, and sense various inputs

In this course, participants will learn how to program using Arduino. Arduino is an electronics platform with easy-to-use hardware and software for making interactive projects. Over the duration of the course, participants will learn to build small electronics projects such as LED night lights, light theremin, memory games and guess the word games. These hands-on projects will be great to introduce individuals to electronics.



Applied Learning



#	Courses	Difficulty	Audience
1	Food Science and Technology	✦✦	Sec/JC
2	Smart Cities	✦✦	Sec/JC
3	Sustainability	✦✦	Sec/JC
4	Healthcare Technology	✦✦	Sec/JC

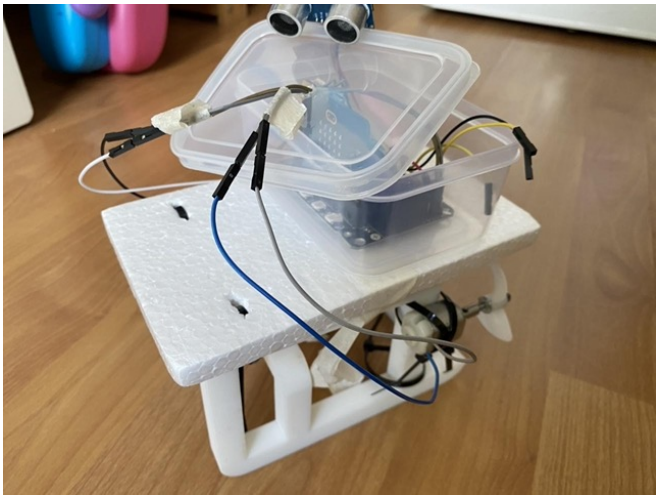
Schools have introduced a wide range of applied learning programmes to help their students connect academic knowledge and skills with real world application. Many such programmes use computing platforms as part of a multi-disciplinary curriculum to engage students.

Over the years, we have accumulated extensive experience building customised programmes that align with the school needs in applied learning settings, whether it is by integrating mobile app development with an exploration of food science, or combining physical computing skills and knowledge to allow students to explore new healthcare technologies.

A majority of the programmes involve co-creation of content with multiple stakeholders, and designing the curriculum for engagement and development of soft skills. It has been a fulfilling experience integrating our programmes with other content experts, seeing how the combination of knowledge and skills from different domains has created an incredible learning journey for students.

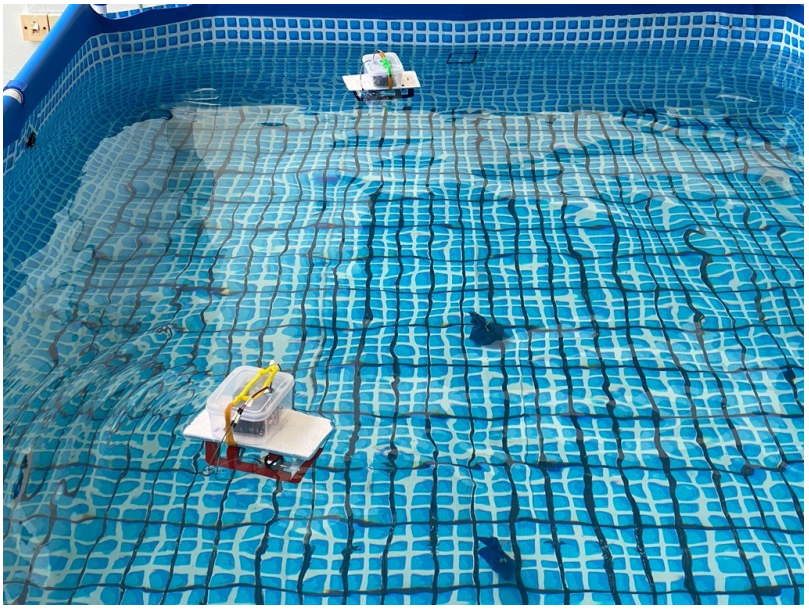
micro:bit Transport Systems

We worked with a school that wanted to incorporate their Applied Learning Programme (ALP) which was on Marine Robotics. They wanted their students to learn about the technologies used in vehicles. We explored land-based and sea-based transport systems, using the Maqueen micro:bit robot to emulate a car, and a motor propelled 'boat' based on the SeaPerch underwater robotics platform.



In the first half of the programme, they learnt how to use and program various external sensors. For instance, they built cars with automatic headlights that would turn on and off when it detects changes in the surrounding light. They also employed distance sensors to know how far vehicles are away from each other.

For the remaining programme, they figured out how controlling motors would affect movement in a water-based environment. Using an inflatable pool, students were able to control and race their boats as well.



In both the land and sea-based transportation systems, they learnt to use the radio built into the micro:bit to control their vehicles remotely.

Applied Learning

1. Food Science & Technology Applied Learning

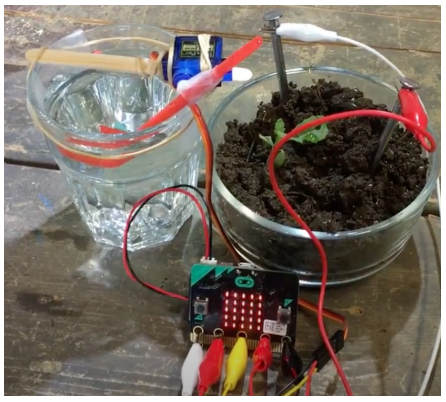
Level:	Beginner
Target group:	Secondary & Up
Duration:	12–24 hrs
Platforms:	Micro:bit, M5Go, Thunkable

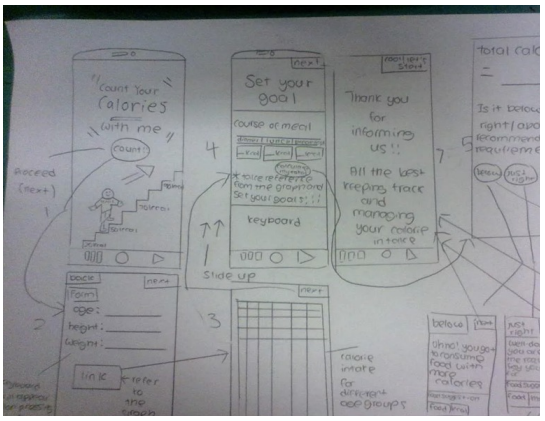
Lesson Outcomes:

- ★ Explore the use and development of recent technology in the agricultural, food processing and food distribution sectors
- ★ Hands-on projects exploring Foodtech and Agritech prototypes

Learners will explore issues in Food Science and/or Agriculture, and relate them to technology-based solutions. In different configurations of this programme, learners have explored agritech to build smart watering systems and weather monitoring stations, or alternatively explored our relationship with consuming food by building mobile apps for ordering food online, educating the public on the cultural diversity of food in different communities, and for helping to screen for dietary restrictions such as allergies.

This programme may include elements of UI/UX design or include the incorporation of design thinking phases to guide students in tackling authentic problem statements and coming up with technology-enabled solutions.





Applied Learning

2. Building Smart Cities Applied Learning

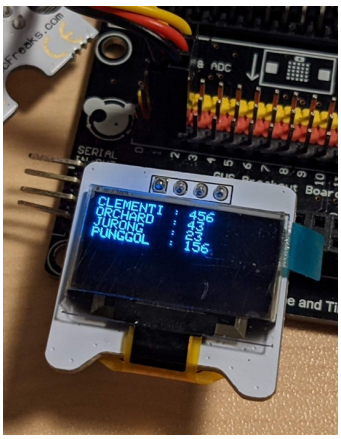
Level:	Beginner
Target group:	Secondary & Up
Duration:	12–24 hrs
Platforms:	Micro:bit

Lesson Outcomes:

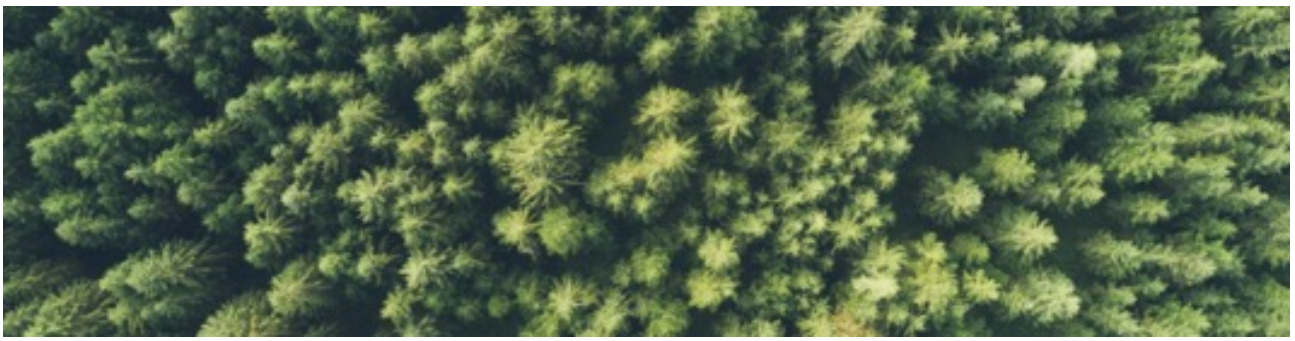
- ★ Explore the evolution of cities, and identify the elements of a Smart City
- ★ Understand the necessity of widespread data collection as a key input to Smart Cities
- ★ Appreciate the privacy considerations arising from data collection
- ★ Hands-on projects exploring Smart City prototypes such as traffic lights and lamp posts

Participants will look at how large scale systems and infrastructure can be, and has been, made smart, in cities such as Masdar, Songdo and Singapore. While we may be quick in adopting the latest gadgets for ourselves, making whole cities and nations smart takes a lot more planning and effort. Participants will be introduced to such planning through a few case studies. They will also be introduced to the micro:bit, and will build their own Smart City (albeit, on a smaller scale) by implementing some of the solutions adopted by existing smart cities on the Micro:bit platform.

The programme may include the incorporation of design thinking phases to guide students in tackling authentic problem statements and coming up with technology-enabled solutions.



3. Sustainability Applied Learning



Level:	Beginner
Target group:	Secondary & Up
Duration:	12–24 hrs
Platforms:	Micro:bit, MakeCode Arcade, Thunkable

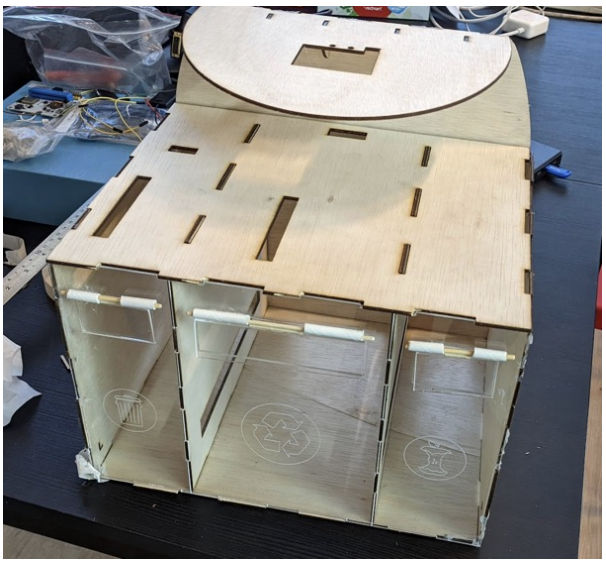
Lesson Outcomes:

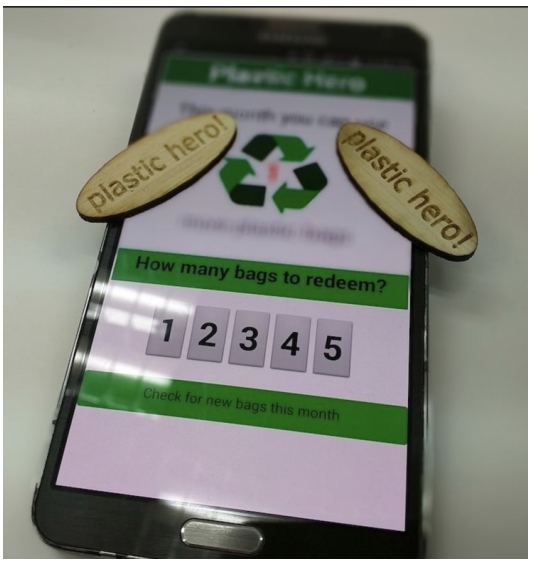
- ★ Increase awareness of environmental and sustainability issues
- ★ Hands-on projects exploring environmental and sustainable prototypes such as smart recycling bins

Learners will explore projects related to environmental sustainability. Environmental sustainability is important to ensure that future generations have the natural resources to live an equal, if not better, way of life as compared to current generations. It can be achieved by responsibly interacting with the planet to maintain the natural resources.

When using the micro:bit, students will create projects such as smart recycling bins that can separate recyclables from non-recyclables. Alternatively, if MakeCode Arcade is chosen, they will build games to spread the message of sustainability. Finally, if using the Thunkable platform, they might build a smartphone app that connects bakeries with excess food to charities that experience food insecurity. The project they build provides an opportunity for them to deeply explore issues of environmentalism and sustainability.

This programme may incorporate design thinking phases to guide students in tackling authentic problem statements and coming up with technology-enabled solutions.





Applied Learning

4. Healthcare Technology Applied Learning

Level:	Beginner
Target group:	Secondary & Up
Duration:	12–24 hrs
Platforms:	Micro:bit

Lesson Outcomes:

- ★ Understand how technology is transforming Healthcare
- ★ Explore the personalisation of Healthcare through smart wearables
- ★ Build smart wearables such as a pedometer and fall detection strap

Technology has impacted so many domains, and healthcare is one such field that has benefited greatly from advancements in wearable and Artificial Intelligence technology. Learners will better understand how these new technologies work, the effort required to apply these technologies in new products and services and the current limitations of these technologies. In an AI-focused version of this course, participants will learn about how AI is transforming healthcare in the areas of smart records, medical imaging and diagnostics, drug discovery, and development and prediction of diseases. In a microcontroller-focused version of this course, they will build their own wearable health-tech prototypes such as a pedometer, zen balance game and fall detection strap. By learning how electronic sensors work to create these health devices, individuals will be better able to appreciate the potential for electronic devices to impact our health.

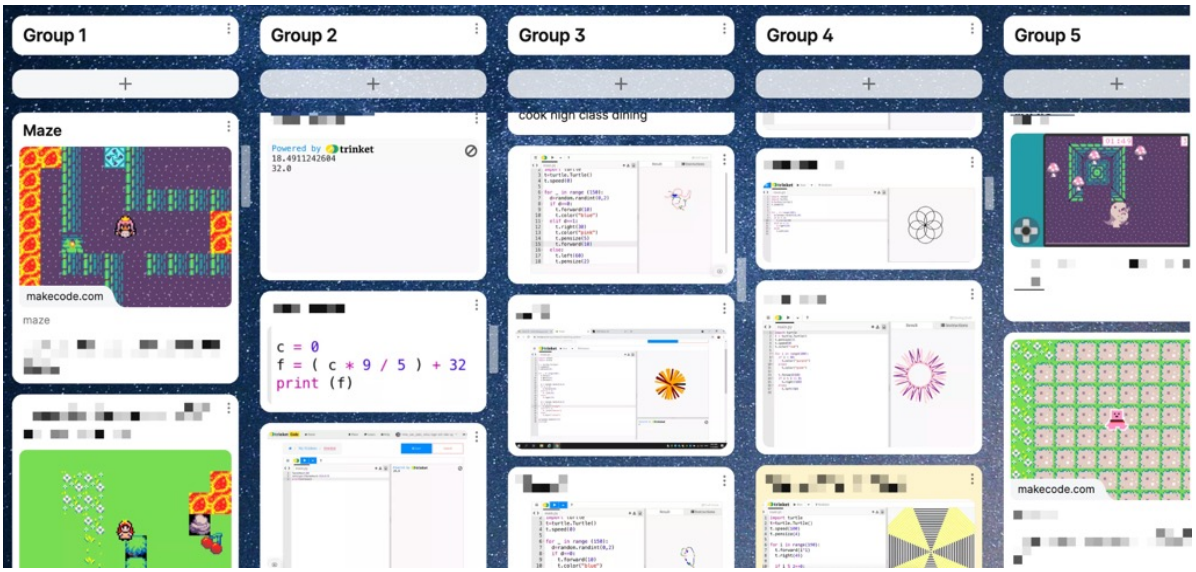
This programme may incorporate design thinking phases to guide students in tackling authentic problem statements and coming up with technology-enabled solutions.

Computer Science

#	Courses	Difficulty	Audience
1	Intro to Computer Science with Minecraft	✦	All Ages
2	Computer Science Foundations with Sphero	✦✦	All Ages
3	Generative Art and Animation with Python	✦✦	Sec/JC
4	Intro to Computer Science with Python	✦✦✦	Sec/JC
5	Python for Business	✦✦✦✦	Sec/JC

Computer Science is a field of study in its own right, and we have designed comprehensive courses that explore computer science concepts in depth. For younger ages, we use platforms such as Minecraft and Sphero so that the codes that youths write create a visible output, whether in the immersive world of Minecraft or the Sphero robotics platform. On the other hand, teenage and adult learners work with the Python programming language, using it to create generative art and animation using the Turtle library, or solve programming tasks from coding competitions.

Computer Studies Programme

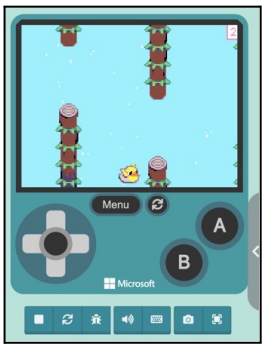


We collaborated with a Singapore secondary school to deliver a 2-year computer studies programme for all their Secondary 1 and 2 students.

During this programme, secondary 1, students picked up coding and computational thinking in a variety of different platforms including LEGO Mindstorms, Scratch and Micro:bit. Meanwhile, secondary 2 students were introduced to Python programming, and were engaged in more formalised Computer Science concepts. They applied their Python programming knowledge and skills to create a plethora of games on the MakeCode Arcade platform as well.

Throughout the delivery of this programme, we shared historical events in computing and discussed current-affairs with the students to increase their ability to think critically about how technology and the evolution of computing would affect them in the future.

We worked closely with the school to develop and refine their computer studies curriculum over multiple years to broaden their students' exposure to the field of Computer Science, enabling them to make an informed decision about choosing their educational pathways, and engaging them with coding challenges that the students find relevant and exciting.



Computer Science

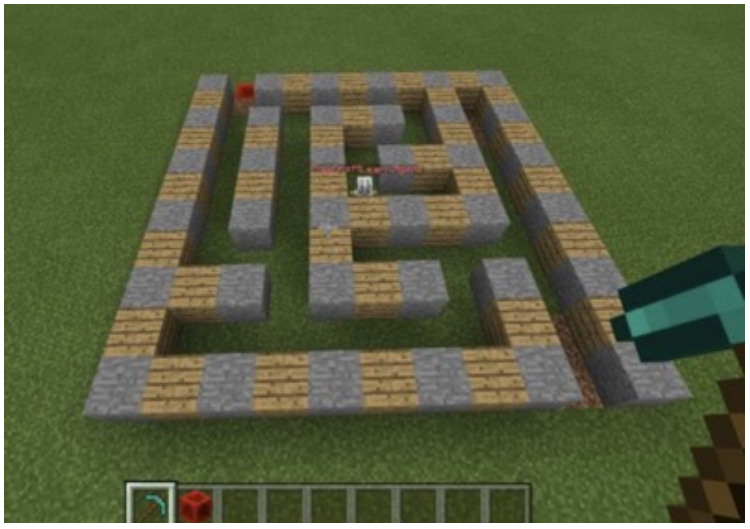
1. Intro to Computer Science with Minecraft

Level:	Beginner
Target group:	Upper Primary & Up
Duration:	4–24 hrs
Requirements:	Minecraft: Education Edition
Prerequisites:	–

Lesson Outcomes:

- ★ Learn computational concepts (e.g., variables, functions)
- ★ Explore block-based, Javascript or Python Programming
- ★ Develop algorithms to automate the building of complex structures (e.g. zoos, mazes etc.)

In this course, participants will learn coding using the Minecraft environment. Minecraft is a highly popular video game and is an open-ended sandbox environment with no clear end goal. Instead, players are encouraged to freely choose how they want to interact in this simulated 3D world, allowing the creation of amazing environments like famous landmarks. This makes Minecraft a perfect environment for teaching computational thinking, especially with its Education Edition which integrates a coding editor that supports block-based Javascript and Python programming languages. In this course, participants will first be introduced to Minecraft and will explore a series of computational concepts e.g., variables, functions etc. using the Minecraft environment.



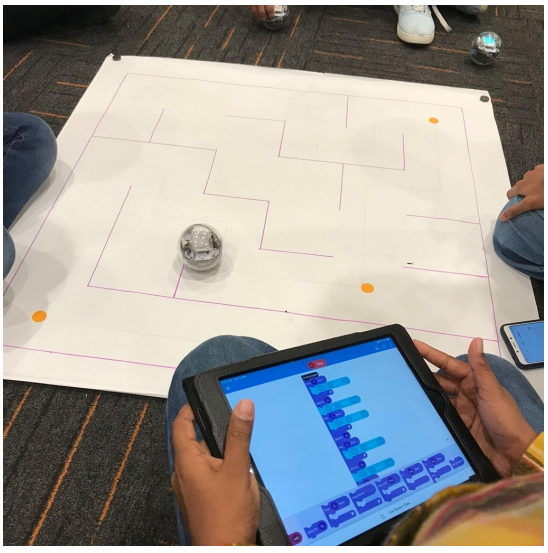
2. Computer Science Foundations with Sphero

Level:	Beginner
Target group:	Lower Primary & Up
Duration:	12 hrs
Requirements:	Sphero Bolt
Prerequisites:	–

Lesson Outcomes:

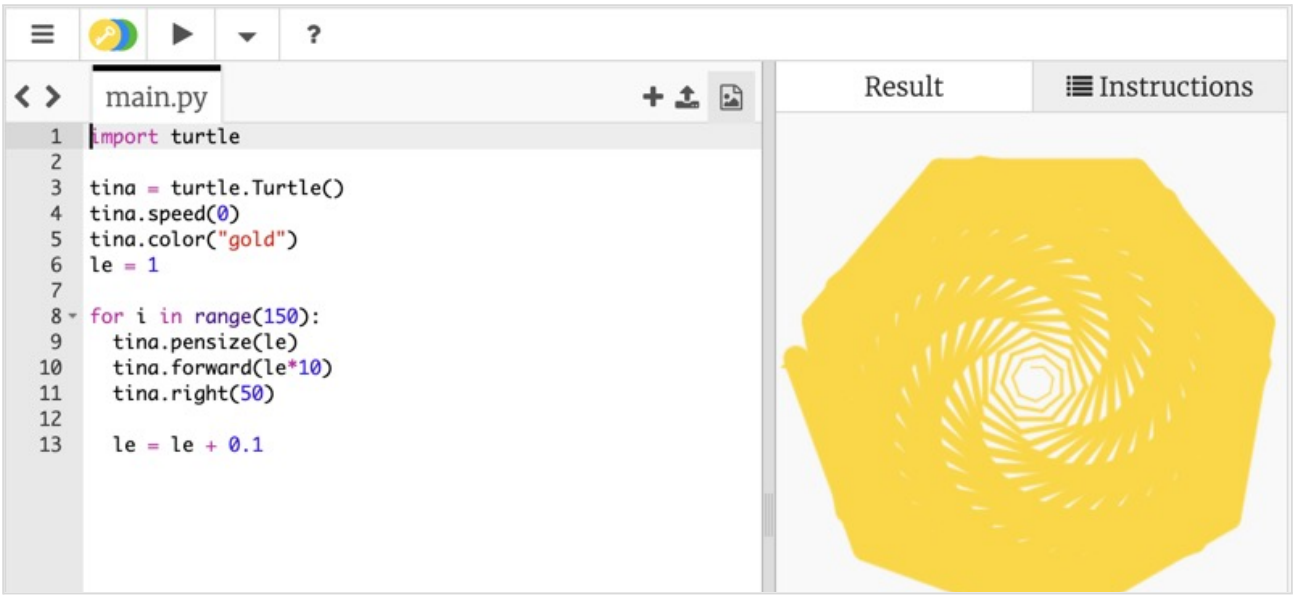
- ★ Learn computational thinking by programming the Sphero robot
- ★ Measure time, speed, distance and other metrics
- ★ Program specific functions such as motion, direction and more!
- ★ Navigate Sphero through terrains using an app or via code

In this course, Sphero will be used to introduce participants to foundational computer science concepts such as algorithms, variables and conditionals while allowing them to practise computational thinking in a project-based, fun and authentic environment. Sphero is a spherical robot toy that can be controlled by a smartphone or tablet. Designed for learner progression, Sphero can be programmed in different ways – by drawing, using block-based coding, or typing in Javascript. This course is great for instilling confidence in individuals when it comes to computer science so that when they encounter these concepts in the future, they will be better equipped to grasp them at a deeper level.



Computer Science

3. Generative Art and Animation with Python

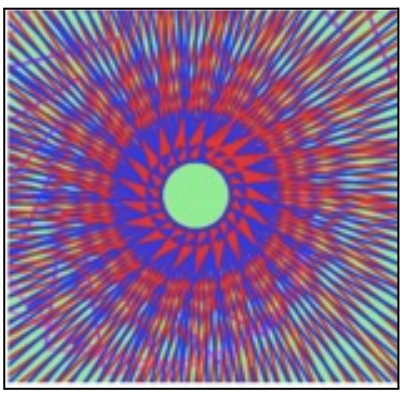
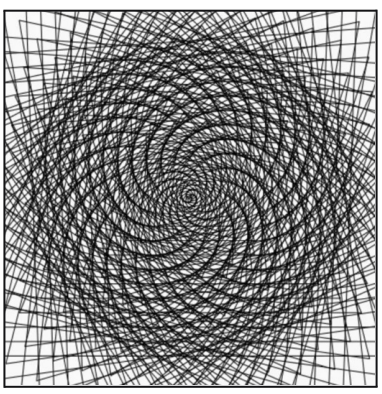


Level:	Beginner
Target group:	Secondary & Up
Duration:	12–24 hrs
Requirements:	–
Prerequisites:	–

Lesson Outcomes:

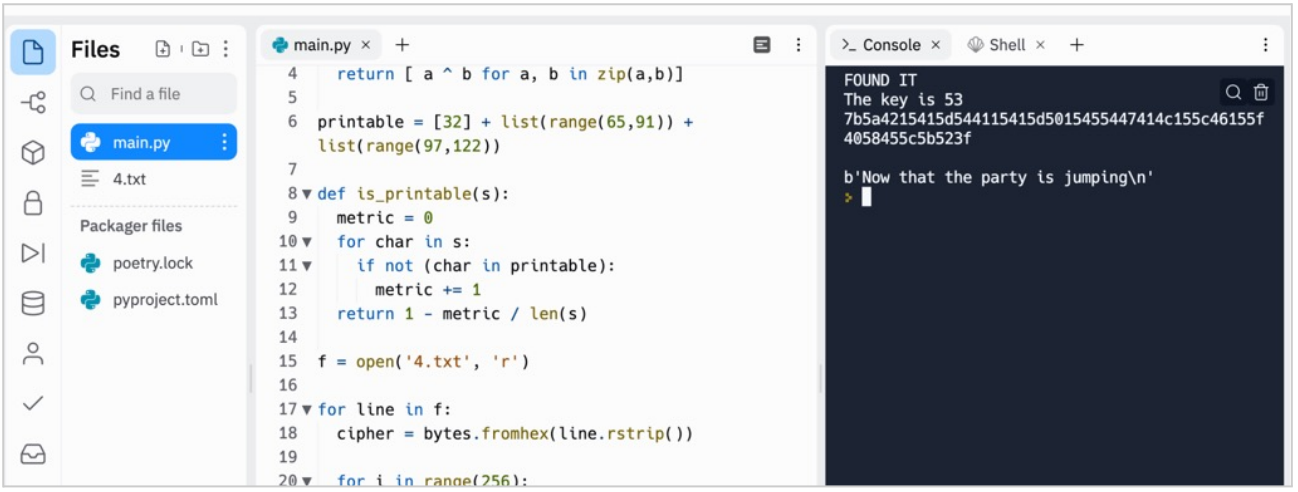
- ★ Understand computational thinking concepts such as variables, operations, loops, and conditionals
- ★ Read and write basic Python code
- ★ Create graphics/animations through code

In this course, participants will learn to create digital art and animations using syntax-based programming in Python. Individuals will be introduced to the Turtle module and the concept of loops and functions in order to explore complex algorithms for creating visually intricate drawings. They will also learn how to create simple animations via the processing development environment by way of the Python mode. Finally, they will learn to implement physical laws in code to create simulations of force and motion to create naturalistic animations of bouncing balls, gravitational attraction and waves. This course will help to improve participants' overall coding abilities as well as their understanding of how to think computationally when implementing algorithms.


Computer Science

4. Intro to Computer Science with Python

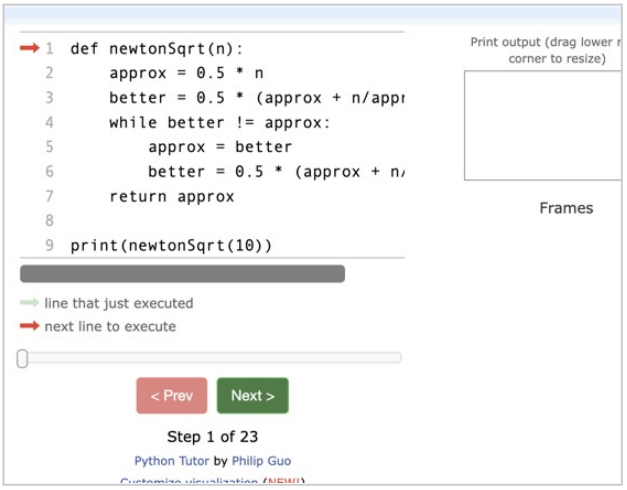


Level:	Intermediate
Target group:	Secondary & Up
Duration:	24 hrs
Requirements:	–
Prerequisites:	–

Lesson Outcomes:

- ★ Understand computational thinking concepts such as variables, operations, loops, and conditionals
- ★ Read and write basic Python code
- ★ Hands-on projects exploring more complex scenarios such as image processing and random drawing

This course serves as a starting point for introductory programming in Python, a powerful and ubiquitous programming language with a gentle learning curve. The curriculum of this course is designed to introduce core computer programming concepts - variables, operators, functions, lists, loops and more - through a variety of platforms and hands-on projects, allowing participants to repeatedly apply what they know in different contexts in order to reinforce their understanding.



Computer Science

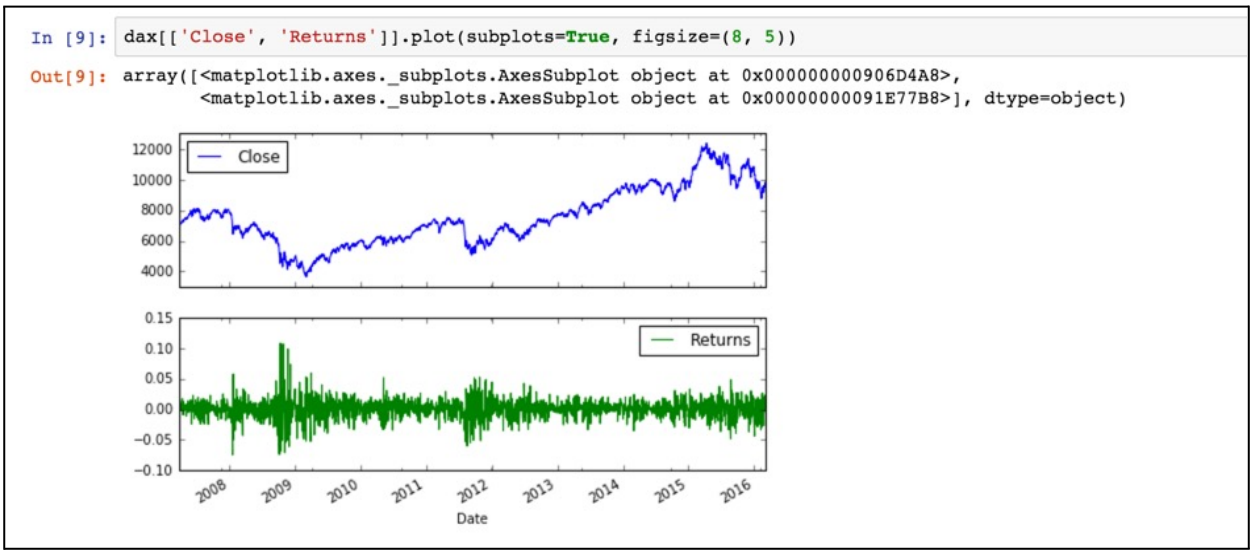
5. Python for Business

Level:	Advanced
Target group:	Adults
Duration:	24 hrs
Requirements:	–
Prerequisites:	–

Lesson Outcomes:

- ★ Read and write Python code
- ★ Consolidate data from reports of varied templates
- ★ Crawl data from e-commerce sites to evaluate an organisation's pricing strategies
- ★ Encrypt documents in the workplace securely

This course will be a fast-paced introduction to Python, the popular programming language, delivered in a context relevant to professionals (for example, accountants, management executives and legal professionals for whom we have previously conducted this course). Python is at once a beginner-friendly language used in introductory Computer Science courses all over the world, while also being powerful and flexible enough to be used in a variety of professional settings e.g. data science, artificial intelligence, and web development. In this course, participants will look at how Python can be used to consolidate data from reports of varying templates, crawl data from e-commerce sites to evaluate an organisation's pricing strategy, and make documents in the workplace secure from prying eyes.

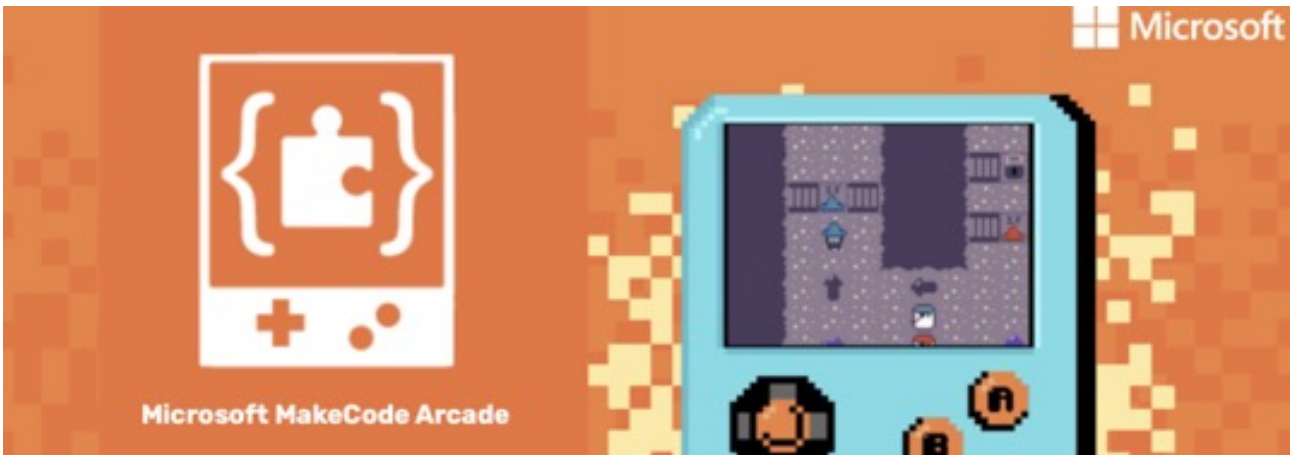


Game Development

#	Courses	Difficulty	Audience
1	Game Development in MakeCode Arcade	✦✦	All Ages
2	Game Development in Unity Bolt	✦✦✦✦	Sec/JC

Everyone loves to play computer games, and it is easy to transfer that natural engagement into having them build their own computer games. While seemingly frivolous to the outside eye, game development requires application of complex computational thinking skills, an understanding of a variety of different coding design patterns, and integration of digital graphics and animation to create an aesthetically-pleasing game. For younger learners and beginners, MakeCode Arcade is a great development environment. Meanwhile, more experienced learners would want to tap on the power of Unity, the same platform that powers immensely popular games like Cities: Skylines, Pokemon and Overcooked.

1. Game Development in MakeCode Arcade

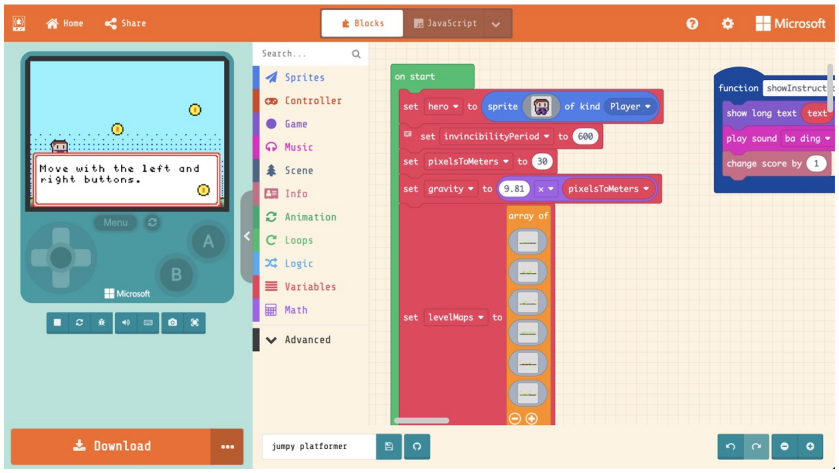


Level:	Beginner
Target group:	Upper Primary & Up
Duration:	12–24 hrs
Requirements:	–
Prerequisites:	–

Lesson Outcomes:

- ★ Appreciate the joy of game development
- ★ Understand core computational thinking concepts such as variables, loops, & conditionals
- ★ Create game prototypes with MakeCode Arcade, playable on browsers

In this course, participants will learn to develop games using the MakeCode Arcade platform. The MakeCode Arcade platform is a beginner-friendly game development platform from Microsoft and is available on any web browser (Windows, Mac, Chromebook, iOS, Android). It focuses on letting users create retro-style games which are simple and engaging for all age-levels - without feeling age-inappropriate for older students, or adults. This course will engage participants in a progressive coding curriculum with a focus on game development. Individuals will be exposed to basic coding concepts and are challenged to use code to solve logic programming problems in order to create their games.



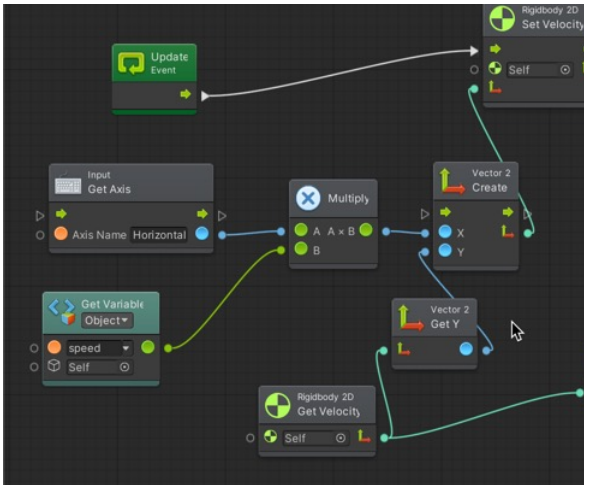
2. Game Development in Unity Bolt

Level:	Advanced
Target group:	Secondary & Up
Duration:	24–36 hrs
Requirements:	–
Prerequisites:	–

Lesson Outcomes:

- ★ Develop a foundation in programming and game development concepts
- ★ Develop curiosity and competency on a modern cross-platform game engine
- ★ Build real-world playable games

In this course, participants will learn to develop games for the Unity game engine in a beginner-friendly, flow-based coding environment. Individuals will be introduced to Unity, a freely available cross-platform game engine, and will develop a series of games in Unity. For example, they will learn the flow-based Bolt programming environment in Unity and will learn how to interface Unity with serial input by working with microcontrollers (e.g. Arduino). This course will help individuals develop a foundation in programming and game development concepts, develop curiosity and competency on a modern cross-platform game engine, and adapt games for use with microcontrollers. We are providing this course as an Authorized Training Partner for Unity.



Web Development

#	Courses	Difficulty	Audience
1	Introduction to Web Technologies (HTML/CSS/JS)	✦✦	All Ages
2	Front-end Web Development with React	✦✦✦✦	Sec/JC

The rapid adoption of the Internet and the World Wide Web in the early 2000s led to massive innovation and disruption to our lives, and gave us the tech behemoths Google, Facebook (now Meta) and Amazon. While web technologies have matured and become more powerful and complex, the basic technologies: HTML, CSS and JS are still the backbone technologies that drive the web browser. For novice learners, we introduce the basic web technologies that will allow them to create their own websites. For more experienced learners, we introduce the popular React framework for rapid web development.

Web Development

1. Introduction to Web Technologies (HTML/CSS/JS)

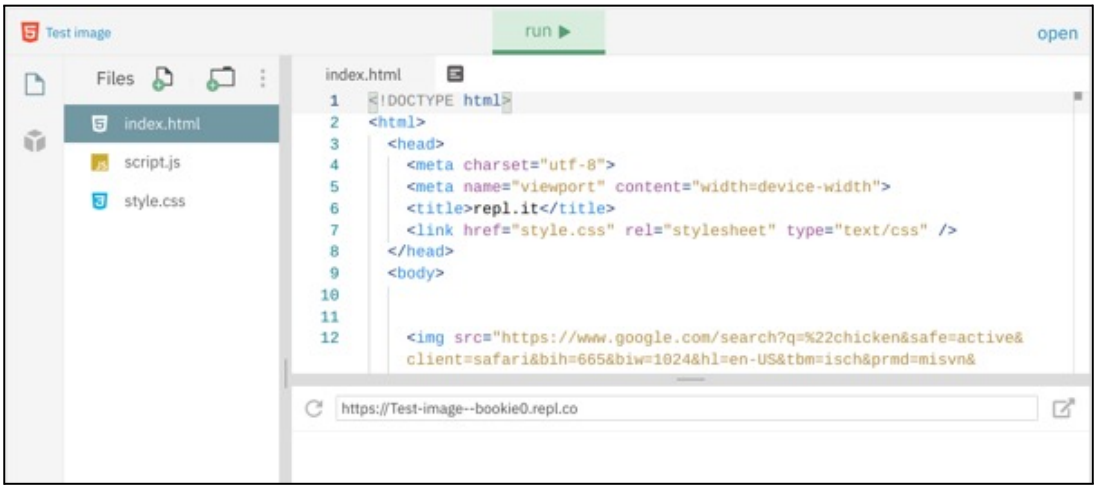
Level:	Beginner
Target group:	Secondary & Up
Duration:	9–12 hrs
Requirements:	–
Prerequisites:	–

Lesson Outcomes:

- ★ Create simple web pages with HTML
- ★ Style web pages with CSS
- ★ Add interactivity and dynamism with Javascript

In this course, participants will learn to develop web applications that run on browsers, using modern HTML, CSS, and JavaScript. We make use of free, powerful online editors such as Replit and Code Sandbox, designed to be collaborative ways to learn web development. By working on projects within the browser, students can jump right into web development with minimal setup.

Hybrid available: This course is available as a hybrid course — we have a full set of recorded lectures for students to self-study, with an instructor joining the class in-person or online to review work and facilitate discussion.



2. Front-end Web Development with React

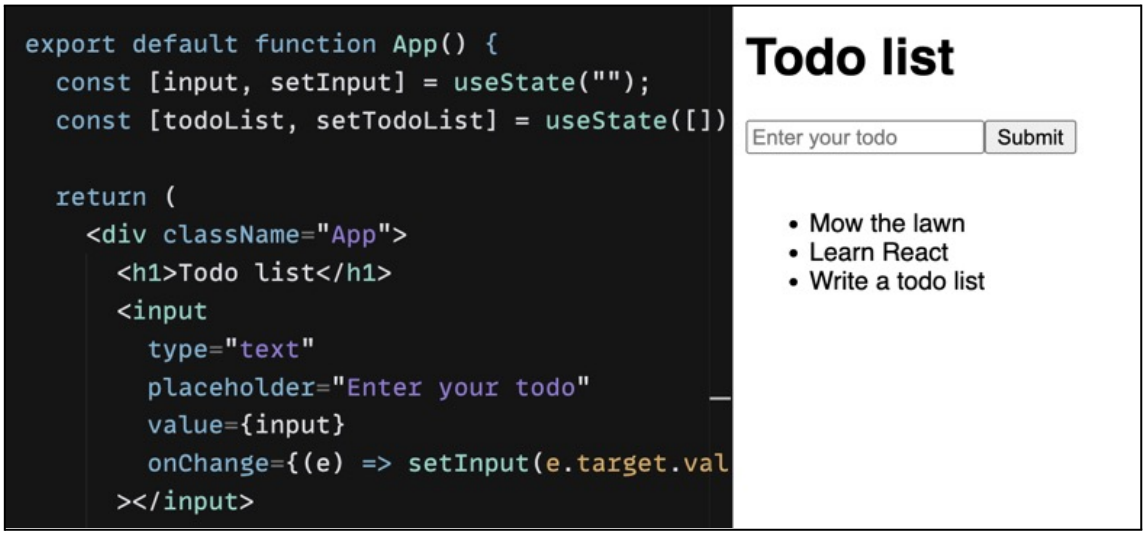
Level:	Advanced
Target group:	
Duration:	24 hrs
Requirements:	–
Prerequisites:	–

Lesson Outcomes:

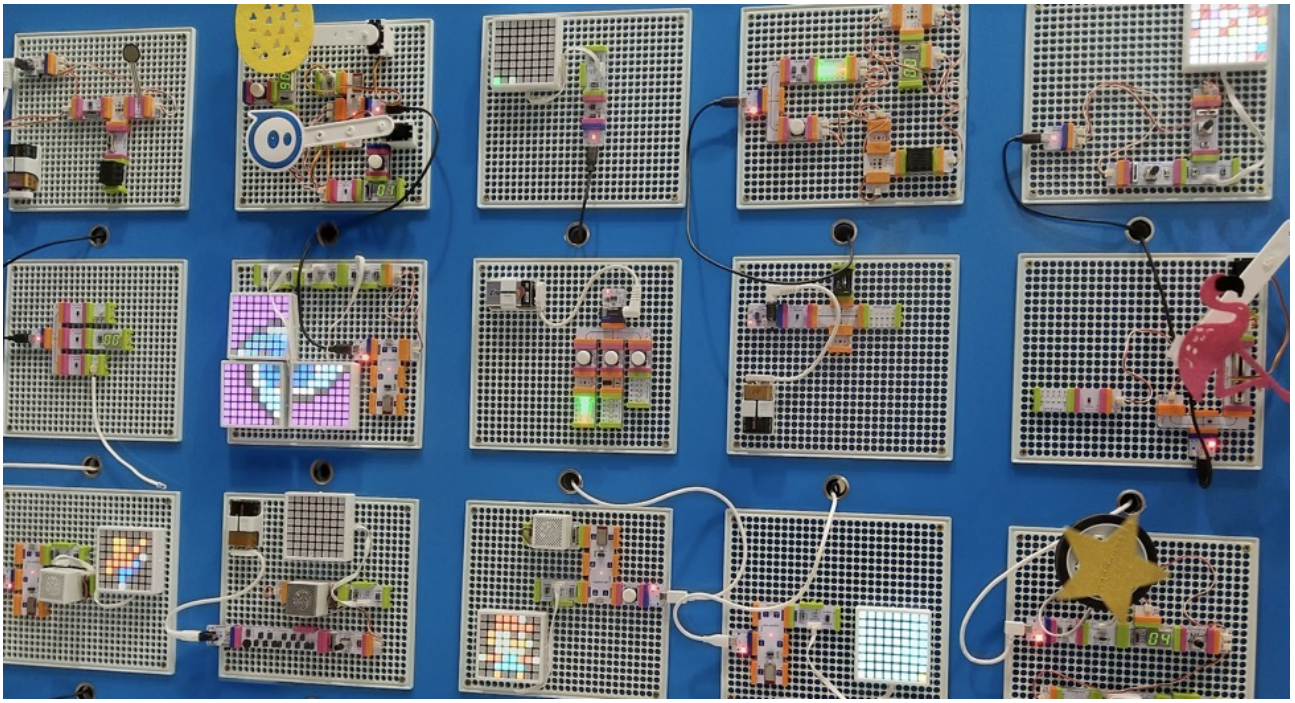
- ★ Implement single-page and multi-page websites in React
- ★ Gain insight on how to use various React features including components and forms
- ★ Learn to create a functional front-end web application using React

Increasingly, everyday apps are powered by web technologies: productivity tools like Google Workspace (Gmail, Google Drive, Google Docs); design tools like Figma; game and streaming platforms like NVidia's GeForce NOW and Netflix; and even desktop apps such as Discord, Visual Studio Code, and Microsoft Teams.

In this course, participants will learn how to build React-based web apps. React is a free and open-source front-end JavaScript library forming the basis of many web apps. After a refresher on HTML, CSS, and introductory JS, students will pick up React and learn how to build web apps using online editors such as Code Sandbox (and, if computers allow, using Visual Studio Code and local NPM installations).



Engineering



#	Courses	Difficulty	Audience
1	Design, Innovation and Creative Engineering	✦✦	Primary

Engineering is the most common undergraduate degree among Fortune 500 CEOs. This is because when you learn how to think like an engineer, you acquire logical thinking, critical analysis and decision-making skills. All of which are greatly needed in today's fast-paced world. We aim to introduce engineering and its related skills to our learners through a beginner-friendly course.

Engineering

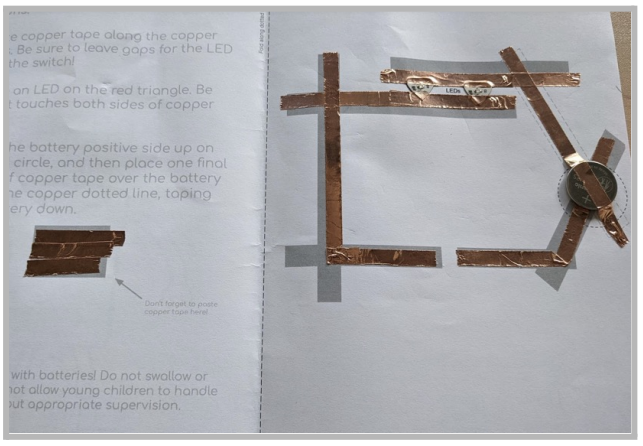
1. Design, Innovation and Creative Engineering

Level:	Beginner
Target group:	Lower Primary & Up
Duration:	12–24 hrs
Requirements:	littleBits, Circuit Stickers
Prerequisites:	–

Lesson Outcomes:

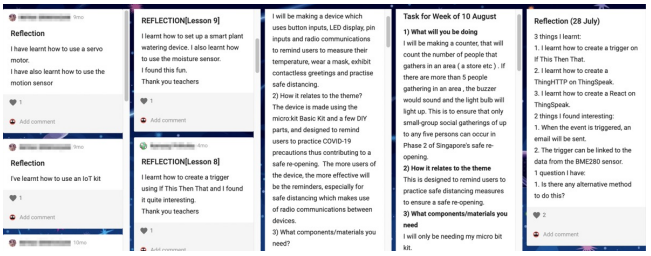
- ★ Hands-on experience with analogue and digital making
- ★ Enhance design thinking
- ★ Tinker and engineer with paper, cardboard and other craft material
- ★ Implement working physical prototypes with littleBits, Circuit Stickers, and other materials

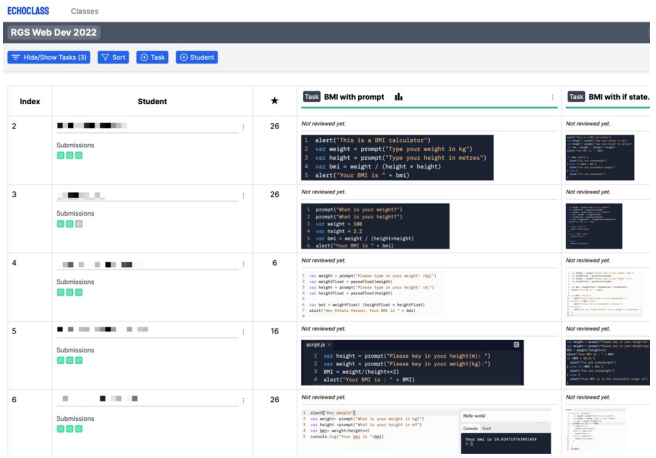
In this course, participants will learn making and design thinking through a series of activities. Individuals will be introduced to digital making, why it is important and what can be made. They will learn about electricity, batteries and circuits using Circuit Stickers and simple LEDs. They will also be introduced to littleBits (input/output, different sensors) and will explore interactive electronics through a series of hands-on activities. Participants will also explore mechanical engineering concepts and principles through building automata, marble machines, and sculptures made from straw, paper and cardboard.



Tools for Teaching & Learning

We've built up over thousands of hours of instructional time every year, — and even built! — a library of useful tools for teaching and learning, which we bring to classrooms as appropriate. These include:





- **Padlet**: We use Padlet as a platform for students to document their learning, and to reflect and share their learnings online with their peers. Reflections are publicly shared within the class to model peer learning, and the students are encouraged to upload photos of their work to serve as a "build log" of their progress.
- **Echo Class**: We commissioned and helped build this open-source tool as a lightweight "classroom management system", where students can join with just a link and a class code, and teachers can track soft-copy submissions easily and effectively. Features include announcements, resources, points, leaderboards, and feedback. z
- **Kahoot**, for quizzes.
- **Discord**, as a moderated online discussion tool. Students can engage in on- and off-topic discussions with one another and with the instructional team, and ask questions in a forum channel. Discord also supports light-weight screen-sharing, great for mini "breakout rooms".
- **Notion**, for collaborative note-taking.
- We are, of course, familiar with all manners of online teaching delivery platforms, such as Zoom, Google Meet, Teams, and Webex.

When working with schools, we are happy to use the school's platforms where appropriate and available, e.g. Teams instead of Discord; Google Classroom instead of Echo Class.

Tinkertanker Pte Ltd
Email: hello@tk.sg
Phone: 69176920

201 Henderson Road #06-19, Apex @ Henderson, Singapore 159545