# Computing

## Year 9

| Assessment Opportunities | Literacy/Reading opportunities | CEIAG Links |
|---|---|---|
| **Within every unit of work, there is a week 3 mini assessment. This is self-marking**<br><br>**There will also be a week 6 assessment which is a mixture of open style examination and the marking of skills the students have learnt e.g. programming.** | **Binary - Images/Sound**<br>**Cyber Security**<br>**Python Programming**<br>**Animation**<br>**Data Science**<br>**Physical Computing** | • Penetration tester<br>• Application analyst<br>• Applications developer<br>• Cyber security analyst<br>• Data analyst<br>• Forensic computer analyst<br>• IT trainer<br>• Machine learning engineer |

Curriculum vision:

"Our aim is to deliver a curriculum that is inclusive, relevant and progressive for all learners."

# Year 9 – Representations: going audiovisual

## Unit introduction

In this unit, learners will focus on digital media such as images and sounds, and discover the binary digits that lie beneath these types of media.

Just like in the previous unit, where learners examined characters and numbers, the ideas that learners need to understand are not really new to them. You will draw on familiar examples of composing images out of individual elements, mixing elementary colours to produce new ones, and taking samples of analogue signals, to illustrate these ideas and bring them together in a coherent narrative.

This unit also has a significant practical aspect. Learners will use relevant software (GIMP and Audacity, in this case) to manipulate images and sounds and get an idea of how the underlying principles of digital representations are applied in real settings.

This unit builds on the material from the Year 8 unit, ‘Representations: from clay to silicon’.

## Overview of lessons

| Lesson | Brief overview | Learning objectives |
|---|---|---|
| 1 Binary mosaic | Digital pictures are formed out of individual pixels (picture elements), just like the Greek and Roman mosaics are formed out of individual pieces of glass or stone. However, unlike their ancient counterparts, the elements in digital mosaics are aligned in rows and columns, with the colour of each element represented as a sequence of binary digits.<br><br>In this lesson, learners will create digital mosaics pixel by pixel, and experience how an image composed of individual coloured elements can correspond to a sequence of binary digits. This will help them form an initial understanding of how the images that they encounter daily in their digital devices translate to nothing more than long strings of bits. | • Describe how digital images are composed of individual elements<br>• Recall that the colour of each picture element is represented using a sequence of binary digits<br>• Define key terms such as ‘pixels’, ‘resolution’, and ‘colour depth’ |

| | | |
|---|---|---|
| | | • Describe how an image can be represented as a sequence of bits |
| 2 A splash of colour | In the early days of personal computers, graphics were displayed in a range of different resolutions and colour depths, depending on the hardware available. Nowadays, while resolution is still being increased, there is no mention of colour depth or the number of possible colours available. We have used 24 or 32 bits for years, as this has been sufficient.<br><br>In the previous lesson, learners were introduced to the idea that the colour of each pixel can be represented as a sequence of binary digits. In this lesson, they will explore the most common representation of colour as a mixture of red, green, and blue: the level of each of these colours in the mixture is represented using an 8-bit sequence, producing a total of 24 bits to represent the colour of any single pixel.<br><br>Learners will also build on their existing knowledge to calculate the representation size of digital images. | • Describe how colour can be represented as a mixture of red, green, and blue, with a sequence of bits representing each colour's intensity<br>• Compute the representation size of a digital image, by multiplying resolution (number of pixels) with colour depth (number of bits used to represent the colour of individual pixels)<br>• Describe the trade-off between representation size and perceived quality for digital images |
| 3 Collage | After introducing learners to the ideas behind digital image representation, it's now time for a hands-on approach. In this lesson, learners will use appropriate software to perform a range of image manipulation functions and complete specific tasks and challenges.<br><br>Learners will already have varying levels of experience and proficiency in using image editing software, so this is a flexible lesson, with a range of | • Perform basic image editing tasks using appropriate software and combine them in order to solve more complex problems requiring image manipulation<br>• Explain how the manipulation of digital images amounts to |

| | | |
|---|---|---|
| | activities provided to suit different needs and tastes. Please note that this lesson is not intended to be a comprehensive introduction to image editing.<br><br>**Week 3 multiple choice quiz will be completed within this class. This will inform the teacher of any misconceptions that need to be addressed before moving on.** | arithmetic operations on their digital representation<br>• Describe and assess the creative benefits and ethical drawbacks of digital manipulation (Education for a Connected World) |
| 4 Good vibrations | Tracing the steps of a hiker through the altitude data that she transmits, learners will familiarise themselves with the basic concepts necessary for understanding any analogue to digital conversion: samples, sampling rate, and sample size.<br><br>The main goal is for learners to understand the 'big picture' of how sound is captured, digitised, manipulated, and reproduced in digital devices. | • Recall that sound is a wave<br>• Explain the function of microphones and speakers as components that capture and generate sound<br>• Define key terms such as 'sample', 'sampling frequency/rate', 'sample size'<br>• Describe how sounds are represented as sequences of bits |
| 5 Sonic playground | First, learners will revisit the digitisation process, in order to understand how the sampling rate and the sample size affect the size and quality of the representation. Next, they will use a sound editing program that will allow them to experiment with sound to complete specific tasks and challenges.<br><br>Learners will have varying levels of experience and proficiency in using sound editing software, so this is a flexible lesson, with a range of activities provided to suit different needs and tastes. | • Calculate representation size for a given digital sound, given its attributes<br>• Explain how attributes such as sampling frequency and sample size affect characteristics such as representation size and perceived quality, and the trade-offs involved |

| | | |
|---|---|---|
| | The instructions in the worksheets are tailored to Audacity (audacityteam.org), which is open-source and cross-platform. However, the tasks can be performed with most sound editing software. | • Perform basic sound editing tasks using appropriate software and combine them in order to solve more complex problems requiring sound manipulation |
| 6 Always another way | **To conclude the unit, learners will spend half the lesson completing a summative assessment.**<br><br>In the time remaining, learners will be introduced to alternative (symbolic) representations for images and sound, such as vector graphics and MIDI music. They will also be introduced to what compression is and why it is necessary. | • Recall that bitmap images and pulse code sound are not the only binary representations of images and sound available<br>• Define 'compression', and describe why it is necessary |

# Keywords

| | |
|---|---|
| BINARY | Number system that computers use to represent data and instructions |
| CHARACTER | Any letter, number or symbol used on a computer |
| REPRESENTATION | How something is stored and shown on a computer |
| PIXEL | One individual dot of colour within an image; sometimes called an element |
| RESOLUTION | The quality of an image indicated by the number of pixels used to create it |
| UNCOMPRESSED | When a file has not been altered in any way and has its original quality |
| MP3 | The most common compressed audio format used today |
| JPEG | The most common compressed image format used today |

| | |
|---|---|
| FILE SIZE | The number of bits used to store a whole file |
| METADATA | Information about the image data that allows a computer to recreate the image from binary |
| BIT RATE | The number of bits used per second for sound/video data |
| ANALOGUE | When data is in its original real-world format |
| DIGITAL | When data has been converted into binary format |
| COLOUR DEPTH | The number of bits used to represent the colour of each individual pixel |
| DENARY | Number system from 0-9 used by humans |
| HEXADECIMAL | A number system with 16 possible options often used to represent colours on the web |
| BIT | The smallest value of binary i.e. 0 or 1 |
| BYTE | 8 bits of data |
| KILOBYTE | Binary unit consisting of 1,024 bytes |
| GIGABYTE | Binary unit consisting of 1,024 megabytes |
| COMPRESSION | The process of making data smaller to save space and increase transmission speed |

# Year 9 - Animations

## Unit introduction

Films, television, computer games, advertising, and architecture have been revolutionised by computer-based 3D modelling and animation. In this unit learners will discover how professionals create 3D animations using the industry-standard software package, Blender. By completing this unit learners will gain a greater understanding of how this important creative field is used to make the media products that we consume. Sessions will take learners through the basics of modelling, texturing, and animating; outputs will include 3D models, short videos, and VR. Links are made throughout to computer science, computational thinking, and the world of work. Tools and techniques learnt in this unit can also be used for 3D printing.

## Overview of lessons

| Lesson | Brief overview | Learning objectives |
|---|---|---|
| Lesson 1: Move, rotate, scale, colour | In this introductory lesson to the unit, learners will look at the impact of 3D animation on the wider world, linking to their own experiences. Learners will be introduced to the basics of making models in Blender: deleting and adding objects; moving, rotating, scaling, and colouring. Learners should finish the lesson having made their own 3D model of a snowman — some will have made a simple snow scene. Links should be made between the naming and reuse of colours, and the computer programming concept of variables. | • Add, delete, and move objects<br>• Scale and rotate objects<br>• Use a material to add colour to objects |
| Lesson 2: Animation, names, parenting | This second lesson covers the basics of keyframe animation, the technique behind how 3D digital animations are made. Learners will be able to explain the differences between keyframing and stop motion animation, and give reasons for why keyframing might be preferable in computer | • Add, move, and delete keyframes to make basic animations |

| | | |
|---|---|---|
| | animation. Learners will gain experience of using the Blender timeline to add, delete, and move keyframes while they animate their own winter scene from the last lesson, or use the template winter scene provided. Finally learners will use naming and parenting to organise their animations. | • Play, pause, and move through the animation using the timeline<br>• Create useful names for objects<br>• Join multiple objects together using parenting |
| Lesson 3: Complex models and colours | This third lesson covers more complex modelling techniques that can be used to build realistic-looking models. Starting from primitive objects, such as cubes and cylinders, learners will use edit mode and the extrude, loop cut, and face editing commands to make a rocket and a chair. Once they have completed their models, they will look at how they can apply different colours to different parts of the same model.<br><br>**Week 3 multiple choice quiz will be completed within this class. This will inform the teacher of any misconceptions that need to be addressed before moving on.** | • Use edit mode and extrude<br>• Use loop cut and face editing<br>• Apply different colours to different parts of the same model |
| Lesson 4: Organic modelling | This fourth lesson covers modelling techniques that are used to make organic/natural-looking models. To do this, learners will first see the importance of breaking symmetry in their models to mimic the real world. The lesson then covers several modelling tools that allow for more natural-looking images, including proportional editing, the knife tool, and subdivision. | • Use proportional editing<br>• Use the knife tool<br>• Use subdivision |
| Lesson 5: Lights, camera, render | This fifth lesson teaches learners how to set up a film shot for rendering. This includes adding extra lighting, adjusting the camera, picking a render mode, and changing the render settings. Learners will understand the range of lights available in Blender, how to set up a camera for a shot, and the benefits and drawbacks of using ray tracing in their films. | • Add and edit set lighting<br>• Set up the camera<br>• Compare different render modes |

| | | |
|---|---|---|
| Lesson 6: Project | This sixth and final lesson brings together all the skills that learners have covered so far. Learners will create a 3–10 second video based on the plan they made for homework after the last lesson. They will self-assess against a set of skills, and ask a peer to assess their work when it is completed.<br><br>**End of unit test will be completed at the end of this term. It will be out of 20.** | • Create a 3–10 second animation<br>• Render out the animation |

# 9 – Cybersecurity

## roduction

akes the learners on an eye-opening journey of discovery about techniques used by cybercriminals to steal data, disrupt sys
te networks. The learners will start by considering the value of their data to organisations and what they might use it for. Th
at social engineering techniques used by cybercriminals to try to trick users into giving away their personal data. The unit wi
common cybercrimes such as hacking, DDoS attacks, and malware, as well as looking at methods to protect ourselves and
gainst these attacks.

## ew of lessons

| | **Brief overview** | **Learning objectives** |
|---|---|---|
| 1: You and your | The aim of this lesson is to introduce the learners to the unit and to help them understand the value of data to companies. The focus will be on what data companies collect from their users and how they use it. Learners will explore this topic through scenarios as well as by looking at the privacy policies of some tech companies that they may already be giving data to. They will be introduced briefly to the law regarding data protection and will reflect on why cybercriminals might want to gain access to data. | • Explain the differen data and informatio<br>• Critique online servi relation to data priva<br>• Identify what happe entered online<br>• Explain the need fo Protection Act |
| 2: Social ring | The aim of this lesson is for learners to become aware of how humans can be a weak point in the system, as well as looking at the social engineering tactics deployed by cybercriminals to dupe users into giving away data that | • Recognise how hun pose security risks t |

| | | |
|---|---|---|
| | could lead to further crime. The lesson starts with the learners using a Scratch program aimed at tricking them into giving away personal information. Learners will then be taken through the common social engineering techniques, completing exercises through the lesson to encourage them to think more deeply about the consequences of the scams and how to avoid becoming a victim. | • Implement strategie minimise the risk of compromised throu error |
| 3: Script kiddies | This lesson allows the learners to explore the concept of hacking and the techniques used by hackers to exploit computer systems. The lesson starts with the learners looking for clues to hack into a friend's account to help his parents find out where he is. They will then be forced to think about the ethics behind their actions. The rest of the lesson looks at terms such as brute force attacks, hacktivists, script kiddies, and DDoS attacks. Some of the key terminology is introduced around the real-life example of the Dyn attack that disabled DNS servers (mostly in the USA) for a time. The lesson will conclude with the learners exploring the Computer Misuse Act and the consequences of hacking.<br><br>**Week 3 multiple choice quiz will be completed within this class. This will inform the teacher of any misconceptions that need to be addressed before moving on**. | • Define hacking in th cyber security<br>• Explain how a DDoS impact users of onli<br>• Identify strategies to chance of a brute fo being successful<br>• Explain the need for Computer Misuse A |
| 4: Rise of the | The purpose of this lesson is to make learners aware of malware and the different categories of malware, as well as understanding how they work and the potential damage they can do. This lesson focuses more on the technical side than on prevention methods, which will be covered in Lesson 5 of this unit. This lesson will start with a pretend scenario of the network | • List the common ma threats<br>• Examine how differe malware causes pro computer systems |

| | | |
|---|---|---|
| | having been infected by ransomware; the learners have to decide what action to take. They will then be introduced to the key terms before being instructed to do a research task to create a fact-based quick read on one type of malware they have learnt about. Towards the end of the lesson, the learners will be introduced to web bots and what task they perform on the internet. They will then be shown how bots are used in conjunction with malware and will be given a scenario that allows them to understand the hidden role of bots and what potential influence they could have on societal issues. | • Question how malic can have an impact issues |
| 5: There's no e 127.0.0.1 | The aim of this lesson is for learners to develop their understanding of the risks that cyberthreats pose to a network, followed by an exploration of some of the more common methods of defending a network against attacks, such as firewalls and anti-malware. The learners will look at the more common threats that exist globally before thinking of the threats at the level of a school network. Learners will discuss methods used by network managers to reduce risk. The homework for this lesson is to write a short report to the head teacher on how to manage the most significant risk to the school network. | • Compare security th against probability a potential impact to organisations<br>• Explain how networ protected from com security threats |
| 6: Under Attack | This is the final lesson in the unit, and the learners are encouraged to reflect on the learning that has taken place throughout the unit before taking an end-of-unit assessment. The learners will be prompted to reflect through a game called Under Attack. Learners will work in groups to plan their defence strategy on a tight budget before cyberattacks start to happen. The use of their budget will be key in determining whether or not they were able to defend the organisation against the attack. Learners will | • Identify the most eff methods to prevent cyberattacks |

| | | |
|---|---|---|
| | then take their end-of-unit assessment and if there is time they will be directed to research the available career choices in cyber-defence.<br><br>**End of unit test will be completed at the end of this term. It will be out of 20.** | |

# words

| | |
|---|---|
| ection Act | UK legal Act regarding the handling of sensitive information such as personal details |
| r Misuse Act | UK legal Act preventing the use of computers for crime or malicious purposes |
| | Software installed without consent |
| cation | Can confirm the identity of a user before they're allowed to access certain pieces of data or features of the program. |
| s | Software designed to find and stop viruses |
| | An update designed to make something better or more secure |
| orse | Malware disguised as something safe. |
| vare | Telling the user their computer is infected to scare them into buying a solution |
| | Keeping things secret from others |
| minal | An online criminal |
| | A con scheme designed to steal money or information from people |
| | Unauthorised copying of computer software/media |

| | |
|---|---|
| twork | Online service that allows easier communication between people, such as Facebook |
| | The use of computers to access data without authorisation |
| n | The purpose of scrambling data so it can be sent securely over networks |
| | Attach themselves to certain files and spread across a system |
| | Self replicating virus |
| | Malware disguised as something safe. |
| gineering | A way of gaining sensitive info by influencing people usually over the phone or phishing emails |
| | When emails or texts are sent pretending to be from a bank or retailer asking you for sensitive info |
| ware | Software designed to find and stop malware |
| | Blocks unauthorised access by examining all data entering and leaving the network |

y of St Nicholas
Rooney

# 9 – Python programming with sequences of data

## roduction

troduces learners to how data can be represented and processed in sequences, such as lists and strings. The lessons cov
of operations on sequences of data, that range from accessing an individual element to manipulating the entire sequence. C
aken so that the selection of problems used in the programming tasks are realistic and engaging: learners will process sola
ook texts, capital cities, leaked passwords, word dictionaries, ECG data, and more.

pedagogical tools are employed throughout the unit, with the most prominent being pair programming, live coding, and wo

7 and 8 Programming units are prerequisites for this unit. It is assumed that learners are already able to write Python progra
essages, receive keyboard input, use simple arithmetic expressions, and control the flow of program execution through sele
ructures.

## ew of lessons

| | **Brief overview** | **Learning objectives** |
|---|---|---|
| up | This introductory lesson serves a double purpose: it reconnects learners with Python, making sure they can read and create simple programs that use selection, and it also takes a step forward, providing a very gentle introduction to lists. | • Write programs that messages, receive input, and use simp expressions in assig statements<br>• Use selection (`if`-e statements) to contr of program executio |

| | | |
|---|---|---|
| | | • Locate and correct<br>syntax errors<br>• Create lists and acc<br>individual list items |
| t | This lesson provides learners with an overview of the operations that are commonly performed on lists: adding, removing, or modifying items; locating or counting occurrences of particular items, etc. Learners are presented with a set of short challenges. They are asked to identify the list operations that would be relevant and apply them to perform the required tasks.<br><br>Through these challenges, learners will indirectly gain a better understanding of the sort of problems where lists might be useful. They also get accustomed to using dot notation for list methods, although this is not the focus of the lesson. | • Perform common op<br>lists or individual ite |
| `ile`, crocodile | This lesson revolves around iteration using `while` loops, offering learners a chance to retrieve and apply relevant knowledge. In the first activities, learners will practise using list operations in iterative contexts.<br><br>Learners will be introduced to the similarities between lists and strings, which will be based on what they already know about operations relating to length, membership, and access to individual characters. The final activity requires them to apply these string operations in an iterative context.<br>**Week 3 multiple choice quiz will be completed within this class. This will inform the teacher of any misconceptions that need to be addressed before moving on.** | • Use iteration (`whil`<br>statements) to contr<br>of program executio<br>• Perform common op<br>lists or individual ite<br>• Perform common op<br>strings or individual |

| | | |
|---|---|---|
| mous `for` | In this lesson, learners will use a `for`-loop to iterate over list items. They will initially study a range of examples — to familiarise themselves with its syntax, use, and mechanics — before moving on to apply what they've learnt to similar tasks.<br><br>The activities involve iterating over lists of real-world textual and numerical data, requiring learners to recall and apply knowledge from the previous lessons.<br><br>The lesson ends with a nod towards using `for` to iterate over the characters of a string, which may come in handy when learners attempt to solve problems independently. | • Use iteration (`for` s to iterate over list ite<br>• Perform common op lists or strings |
| a thing | IIn this lesson, learners will be provided with a selection of meaningful mini-projects that will allow them to apply the knowledge and skills they have acquired so far. Each project contains a short introduction that provides context, a detailed description of what learners are expected to develop, and a set of clues that will support them in putting together a solution. Each learner is expected to select one of the mini-projects and complete it within this lesson, or in the first part of the next one.<br><br>Before starting work on the projects, two short activities will provide learners with additional support around accumulating sums and using `for` to iterate over strings. This is generally important and will also prove useful in some of the projects. | • Use iteration (`for` lo iterate over lists and<br>• Use variables to kee counts and sums<br>• Combine key progra language features t solutions to meanin problems |
| up | In this final lesson, learners will be given the opportunity to complete their mini-project or explore a second one. **They will then take a quiz that will assess their grasp of the programming concepts they have encountered throughout the unit.** | |

| | | |
|---|---|---|
| | An optional activity is also provided, for learners that finish early with their assessment quiz, or are simply keen on an additional challenge. | |

# words

| | |
|---|---|
| ates | Values used to state the location of a character or item on the screen |
| | A value stored in memory, that be changed |
| e | Carrying out instructions in order |
| | Making a choice or decision in a program |
| | Repeating instructions or code |
| n | A set of steps to solve a problem |
| on Operator | A character used to compare two values e.g. > or = |
| | A group of variables used in a program |
| sition | Breaking down a task into smaller chunks |
| g | Removing errors from a program |
| | A computer graphic which may be moved on screen |

y of St Nicholas
Rooney

# 9 – Data science

## roduction

, learners will be introduced to data science, and by the end of the unit they will be empowered by knowing how to use data
problems and make changes to the world around them. Learners will be exposed to both global and local data sets and ga
ding of how visualising data can help with the process of identifying patterns and trends. Towards the end of the unit, the le
the steps of the investigative cycle to try to solve a problem in the school using data.

## ew of lessons

| | Brief overview | Learning objectives |
|---|---|---|
| : Delving into ence | The aim of this lesson is to introduce the learners to data science, and in particular, how visualising data can help us to provide insights that may not be as obvious when looking at raw data. The learners will investigate a couple of historical examples that highlight the value in visualising data, before using an online tool to help them visualise a small data set of TV viewing figures in order to gain an insight. The lesson will conclude with learners looking at a data set and deciding what it would be useful to visualise. The homework is then to visualise that data. | • Define data science<br>• Explain how visualis can help identify pa trends in order to he insights<br>• Use an appropriate tool to visualise data look for patterns or |

| | | |
|---|---|---|
| 2: Global data | The previous lesson gave learners an introductory level of understanding of the purpose of data science. In this lesson, they will gain a better understanding of how ever-improving advances in technology have made it more feasible to collect, store, and analyse much larger data sets than previously. The learners will look at global data sets, make predictions, and use visualisations of the global data to prove or disprove their predictions, as well as to investigate anomalies and outliers in the data. | • Recognise example large data sets are life<br>• Select criteria and u to investigate predic<br>• Evaluate findings to arguments for or ag prediction |
| 3: Statistical mind | The focus of this lesson is to introduce the learners to the investigative cycle PPDAC (problem, plan, data, analyse, conclusion) and apply part of this cycle to a data set about roller coasters. The learners start this lesson where they left off, by analysing a graph from a world data set. The graph will be used to highlight the correlations in the data and to investigate outlying data. After being introduced to PPDAC, the learners will be given a scenario to investigate what would make a cool roller coaster. They will refine the problem into questions they can investigate, visualise the data, analyse, and report on their findings.<br><br>**Week 3 multiple choice quiz will be completed within this class. This will inform the teacher of any misconceptions that need to be addressed before moving on.** | • Define the terms 'co and 'outliers' in rela trends<br>• Identify the steps of investigative cycle<br>• Solve a problem by implementing steps investigative cycle c<br>• Use findings to supp recommendation |
| 4: Data for | In this lesson, the learners will develop their understanding of the investigative cycle by investigating a problem themselves. They will do this by investigating the problem of litter in their school. They will work through the first two steps of the cycle (problem and plan). To do this, they will pose questions and think about what data they will need to answer those | • Identify the steps of investigative cycle<br>• Identify the data nee answer a question c the learner |

| | | |
|---|---|---|
| | questions. Learners will then make an electronic data capture form, on which they will go on to enter the data that they have collected. Following this lesson, the learners will work through the remaining steps of the cycle to complete the investigation of the problem. | • Create a data captu |
| 5: Clean it up | In this lesson, learners will continue to develop their understanding of the investigative cycle by working through the data and analysis steps of the PPDAC cycle, using their own problem and the data that they are investigating. The lesson starts with a practical exercise that gives the learners some experience of data cleansing, to help them understand the problems that inaccurate data can pose for data analysis. The learners will then download the data they have collected and clean it before uploading it to CODAP, where they will analyse it further by creating visualisations. In this lesson, the learners will start to find some answers to the questions that they posed previously. It will also act as a platform for them to start drawing the conclusions they need to draw in the next and final lesson of the unit. | • Describe the need f cleansing<br>• Apply data cleansin techniques to a data<br>• Visualise a data set |
| 6: Make a | In this lesson, the learners will complete their school litter project by working through the final steps of the PPDAC cycle (analysis and conclusions). The lesson begins with the learners looking at an example visualisation. They will be encouraged to think about what they can learn from the data, as well as what additional information would be helpful for them to know. This will model the thought process they need to go through when analysing and concluding their projects. The learners will take an end-of-unit assessment before thinking about how they could apply what they have learnt in a context that is relevant to them and their lives.<br><br>**End of unit test will be completed at the end of this term. It will be out of 20.** | • Visualise a data set<br>• Analyse visualisatio identify patterns, tre outliers<br>• Draw conclusions a findings |

# words

| | |
|---|---|
| | Raw data like numerical values and text without context |
| on | Data with added meaning/context |
| | Removing errors and making data readible |
| | Appropriate for the situation |
| | useful for the scenario |
| | Mistakes in the data |
| | Looking at something carefully to carry out an investigation |
| ion | A visual representation that communicates relationships among the data |
| n | To make a final judgement on something |
| tion | To anaylse and scrutinise data to try and find meaning |
| nce | extracting meaning from large data sets in order to provide insights to support decision-making |
| g | To take things out of a set of data |
| tation | What something is showing |
| ic | A collection of graphics with  extra context for the inexperienced audience |
| alisation | A single graph or chart with not much added explanation or context |
| on | The relationship between two or more variables |
| | Compares 2 or more sets of data and has 2 axis |
| | Compares data with visual sections such as the sections of the pie chart |
| tive Cycle | The order of investigation of data - Problem, plan, data, analysis, conclusion |
| | To arraneg data in a certain way e.g. alphabetically |
| | To remove unwanted data temporarily to focus on something specific |

| | |
|---|---|
| **ng** | To change the appearance of a document |
| | A chart showing data in slices of a pie to represent their percentage |
| | A chart to compare items easily by the height of the bars |
| **h** | A graph usually used to compare two variables e.g. cases over time |

# 9 – Physical computing

## roduction

pplies and enhances the learners' programming skills in a new engaging context: physical computing, using the BBC micro

half of the unit, learners will get acquainted with the host of components built into the micro:bit, and write simple programs t
ponents to interact with the physical world. In the process, they will refresh their Python programming skills and encounter a
ing patterns that arise frequently in physical computing applications.

ond half, learners will work in pairs to build a physical computing project. They will be required to select and design their pro
ly, apply what they have learnt by building a prototype, and keep a structured diary throughout the process.

3 and 9 programming units are prerequisites for this unit. It is assumed that learners are already able to write Python progra
les and data structures to keep track of information. They are also expected to be able to combine sequence, selection, iter
ethod calls to control the flow of program execution.

## ew of lessons

| | **Brief overview** | **Learning objectives** |
|---|---|---|
| hysical world | This introductory lesson is meant to get learners acquainted with the micro:bit.<br><br>They will explore its hardware components, so that they develop an awareness of its capabilities. They will also write and execute their first Python programs on the micro:bit, so that they familiarise themselves with the development environment, the practicalities of flashing their programs, and some simple coding patterns. | • Describe what the m<br>• List the micro:bit's in output devices<br>• Use a development environment to write and debug a Python for the micro:bit |

| | | |
|---|---|---|
| | At the end of the lesson, the learners will discuss what makes physical computing different from what they have been doing so far. | |
| ones | Through the course of this lesson, learners will write programs that use the micro:bit's 5×5 LED display for output and some of the built-in sensors for obtaining input. This simple 'bare bones' setup will allow them to focus on the code and the patterns that often arise in physical computing applications. At the same time, they will get the chance to revisit some elementary programming constructs they learnt in previous units.<br><br>At the end of the lesson, learners will be asked if they have had any project ideas while exploring the micro:bit. Designing and building their own project is the ultimate goal of the unit. | • Write programs that micro:bit's built-in in output devices |
| ctions | This lesson provides learners with examples of using the micro:bit's General-Purpose Input Output (GPIO) pins to connect it to external hardware components, such as switches, speakers, and LEDs. The ability to connect the micro:bit to additional components enhances the built-in capabilities for input and output, which extends the range of projects the learners will be able to build.<br><br>The lesson also demonstrates the use of the micro:bit's radio antenna in order to transmit and receive messages wirelessly. This is one of its most versatile capabilities and opens the way for projects that involve multiple micro:bits working together.<br><br>At the end of the lesson, learners will again be asked about their project ideas. This time, they will also be asked to put their ideas on paper as homework, as they will find themselves taking their first creative design steps in the next lesson. | • Write programs that pins to generate out receive input<br>• Write programs that communicate with o devices by sending receiving messages |

| | | |
|---|---|---|
| | **Week 3 multiple choice quiz will be completed within this class. This will inform the teacher of any misconceptions that need to be addressed before moving on.** | |
| it up | The first three lessons allowed learners to explore the individual physical computing components at their disposal. Starting with this lesson, they will build their own physical computing project, thus bringing together what they have learnt into a meaningful creation. | • Design a physical c artifact purposefully mind the problem at needs of the audien and the available re<br>• Decompose the fun a physical computin into simpler features |
| up | The bulk of this lesson is dedicated to developing the learner projects. In pairs, they will work on their project prototype, following the proposal they drafted in the previous lesson. Halfway through the lesson, learners will pause to receive peer feedback, evaluate it, and fill in their project diary. By the end of the lesson, the project prototypes should largely be implemented. | • Implement a physic computing project, v following, revising, a the project plan |
| t up | In this final lesson, learners will add the finishing touches to their projects; they will proceed to document what they have produced and reflect on the journey. Their projects will be evaluated using a rubric, and they will also take a quiz to assess the knowledge and skills they have individually acquired over the course of the unit. The lesson will conclude with a look at other existing physical computing platforms. | • Implement a physic computing project, v following, revising, a the project plan |

| | | |
|---|---|---|
| | **End of unit test will be completed at the end of this term. It will be out of 20.** | |

# words

| | |
|---|---|
| CODE | A simplified programming language, used to describe algorithms |
| N | Where a group of statements is executed repeatedly |
| ART | A visual method of representing an algorithm |
| HM | A set of steps designed to solve a problem |
| CE | Where a set of instructions is executed in order |
| N | Where an option of statements is provided |
| E | A value stored in memory that can change while a program runs |
| | Data type used for positive and negative whole numbers |
| I | A data type that only stores one of two values |
| | Data type that can be used to store any combination of alphanumeric data |
| NT | A value stored in memory that cannot change while a program runs |
| | Numerical data which contains decimal numbers |
| | A type of error that occurs due to incorrect spelling or grammar in a program |
| TER | A data type used for storing a single alphabetic or numeric character |
| OR | Part of the instruction that tells the CPU what to do |
| NTS | Useful for explaining what your code is doing |

| | |
|---|---|
| TION | Used to separate different statements in a program |
| DLING | Reading from and writing to files |
| ent | Used to only complete a process when a condition is true |
| | A mistake in a program |
| | Figure out what a program will do |
| TATEMENT | A set of instructions that will only run based on 2 different conditions |
| OSITION | Breaking down a problem into smaller chunks |