

# INSPIRE COMPUTING

*International*

Student Book

**YEAR 8**



Pearson

# **INSPIRE** **COMPUTING**

*International*

Student Book **YEAR 8**

**Paul Clowrey**



Pearson

Published by Pearson Education Limited, 80 Strand, London, WC2R 0RL  
www.pearson.com/international-schools

Copies of official specifications for all Pearson Edexcel qualifications may be found on the website:  
<https://qualifications.pearson.com>

Text © Pearson Education Limited 2022  
Project managed and edited by Just Content  
Designed and typeset by PDQ Digital Media Solutions Ltd  
Picture research by Integra  
Original illustrations © Pearson Education Limited 2022  
Cover design © Pearson Education Limited 2022  
Cover illustration @ Beehive/Andrew Pagram

The right of Paul Clowrey to be identified as the author of this work has been asserted by him in accordance with the Copyright, Designs and Patents Act 1988.

First published 2022

24 23 22  
10 9 8 7 6 5 4 3 2 1

British Library Cataloguing in Publication Data  
A catalogue record for this book is available from the British Library

ISBN 978 1 292 40428 8

#### Copyright notice

All rights reserved. No part of this publication may be reproduced in any form or by any means (including photocopying or storing it in any medium by electronic means and whether or not transiently or incidentally to some other use of this publication) without the written permission of the copyright owner, except in accordance with the provisions of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency, 5th Floor, Shackleton House, 4 Battlebridge Lane, London, SE1 2HX ([www.cla.co.uk](http://www.cla.co.uk)).

Applications for the copyright owner's written permission should be addressed to the publisher.

Printed in Slovakia by Neografia

The author and publisher would like to thank the following individuals and organisations for permission to reproduce photographs, illustrations, and text:

KEY (t – top, c – center, b – bottom, tl – top left, tr – top right)

**Alamy Stock Photo:** Marek Uliasz 85BR; **Getty Images:** Jasmin Merdan/Moment 2, JakeOlimb/DigitalVision Vectors 24, Muriel de Seze/DigitalVision 46, Jonathan Kitchen/The Image Bank 74, Peter Cade/Stone 94, Eggy Sayoga/Moment 116; **Microsoft Corporation:** © Used with Permission from Microsoft 78, 79, 82, 83; **Paul Clowrey:** 9tl1, 9tl2, 9tr1, 9tr2; **Python Software Foundation:** Python Software 33, 39, 113; **Scratch:** Scratch is a project of the Scratch Foundation in collaboration with the Lifelong Kindergarten group at the MIT Media Lab. It is available for free at <https://scratch.mit.edu>, Creative Commons Attribution-ShareAlike license" 32, 38; **Shutterstock:** Lane V. Erickson vii, Hrecheniuk Oleksii 7, Dikas Studio 8cr1, PotapovAlexandr 8cr2, Pavel Talashov 8br, Maksym93 9cl, Metamorworks 10, Dipu Thakuria 11, Popicon 12, Brian A Jackson 14, Kobchok Ratanavimol 16, Robtek 16, Framesira 17, Vector Department 21, Elenabsl 22, Brian A Jackson 26CR, Jirik V 26br, Jacob Lund 28, Krakenimages.com 31, Asharkyu 34, Shutterstock 36, Allahfoto 49, Musakargo 52, 200dgr 54, Rawpixel.com 60, Trong Nguyen 63, Morten B 64, ARIMAG 67, Black Salmon 68, Nicostock 72, Re2deer 84cr, Vectorrex 84BR, Anna Klepatckaya 85tr, Samby 85cr, Antonio Guillem 90cr, Rawpixel.com 90BR, Rido 92, I000s\_pixels 100, Gorodenkoff 102, Fer Gregory 105, Ifoodijourney 107, Mohdizuan 112, Stokkete 122, RoSonic 123cr1, Ronstik 123cr2, Aslysun 123cr3, Proxima Studio 123cr4, Frame Stock Footage 128, LvNL 132, Inspiring 135, Olivier Le Moal 138; **123RF:** Karel Joseph Noppe Brooks 58, Michal Bednarek 86;

All other images © Pearson Education

# Overview contents

Detailed contents	iv
About the Student Book	vi
<b>Unit 1: Binary and online graphics</b>	<b>2</b>
<b>Unit 2: Data on the Web</b>	<b>24</b>
<b>Unit 3: Processing Data</b>	<b>46</b>
<b>Unit 4: Analysing Data</b>	<b>74</b>
<b>Unit 5: Programming</b>	<b>94</b>
<b>Unit 6: Software and communication</b>	<b>116</b>
<b>Glossary</b>	<b>140</b>

# Detailed contents

## Unit 1: Binary and online graphics

1 Vector graphics .....	4
2 Creating a vector graphic .....	6
3 Bitmaps .....	8
4 Comparing vectors and bitmaps.....	10
5 Binary and pixels .....	12
Mid-unit assessment .....	14
6 Colour depth and binary representation .....	16
7 and 8 Digital image properties .....	18
9 & 10 Binary addition .....	20
End-of-unit assessment .....	22

## Unit 2: Data on the Web

1 Bits, bytes and nibbles .....	26
2 Storage capacity .....	28
3 Binary and Denary .....	30
4 Creating a 'power of' calculator .....	32
5 Internet Bandwidth .....	34
Mid-unit assessment .....	36

6 Creating a bandwidth calculator .....	38
7 & 8 Compression .....	40
9 & 10 Run Length Encoding .....	42
End-of-unit assessment .....	45

## Unit 3: Processing Data

1 Computer hardware .....	48
2 Wearable technology .....	50
3 Logo design .....	52
4 Introduction to spreadsheets .....	54
5 Spreadsheet formulas and functions ...	56
Mid-unit assessment .....	58
6 Spreadsheet modelling .....	60
7 Environmental impact of technology .....	64
8 Presenting data in an informative way .....	66
9 Surveys and questionnaires .....	68
10 Advanced spreadsheet functions .....	70
End-of-unit assessment .....	72

## Unit 4: Analysing Data

1 Social media .....	76
2 What makes a good spreadsheet .....	78
3 Knowledge, data and information .....	80
4 Sorting and filtering a spreadsheet .....	82
5 & 6 Design techniques for an online advertisement .....	84
Mid-unit assessment .....	86
7 and 8 What makes a good chart? .....	88
9 and 10 Designing an infographic .....	90
End-of-unit assessment .....	92

## Unit 5: Programming

1 and 2 Algorithms and turtle programming .....	96
3 and 4 Sequencing and iteration algorithms .....	98
5 Programming errors .....	100
Mid-unit assessment .....	102
6 Using lists in a Python program .....	104
7 Use of selection in an algorithm .....	106
8 Creating a simple chatbot .....	108

9 Subprograms in an algorithm .....	110
10 Creating a missing word game .....	112
End-of-unit assessment .....	115

## Unit 6: Software and communication

1 Surveillance technology .....	118
2 Types of software .....	120
3 Operating systems .....	122
4 Utility software .....	124
5 Internet hardware .....	126
Mid-unit assessment .....	128
6 Wireless communication technology .....	130
7 The internet and the World Wide Web .....	132
8 and 9 Surveillance ethics .....	134
10 Data breaches and cyber-crime .....	136
End-of-unit assessment .....	138

Glossary .....	140
----------------	-----

# Welcome to Inspire Computing

Whether for school, fun, work or staying in touch with relatives around the world, digital technology is all around us.

Through coverage of ICT and Computer Science you will discover how this amazing technology works, how it connects the world together and it has revolutionised the classroom, workplace, and home.

## Related topics

Other topics linked to the subject that can also be explored.

## Real-world examples

How the learning applies to the world outside the classroom.

**Sorting algorithms**

**Related topics**

- Understanding and using variables
- Real-world systems

**Key words**

- ascending
- descending
- list
- numerical
- program
- sort
- string

**Learning objectives**

- Understand the purpose of a sorting algorithm.
- Create, run and edit a simple sorting program.

**A simple sort**

A sorting algorithm orders a list of values into the order required. Most sorts are numerical or alphabetical. Here are two examples of sorts.

Numerical sort	Alphabetical sort
Original values: 1,6,4,9,1,3	Original values: 1,0,2,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0
Sorted into ascending order: 1,2,3,4,6,9	Sorted into ascending order: A,D,E,P,H,Z
Sorted into descending order: 9,6,4,3,2,1	Sorted into descending order: Z,U,P,E,D,A

**Real-world examples**

Sorting algorithms are used for:

- high score tables in a computer or smartphone game
- the songs in your playlist
- the contacts in your smartphone
- sales of items in a shop, sorted by price or product.

**A sorting program**

This program demonstrates how to create a simple alphabetical sort using Python. The `sort()` command is applied to the characters stored in the `letters` list.

```
letters = ['B', 'T', 'S', 'P', 'M', 'W', 'A', 'C']
letters.sort()
print(letters)
```

You can create a numerical sort in the same way:

```
numbers = [54,5,98,1,45,68]
numbers.sort()
print(numbers)
```

## Key words

These are important words to know!

## Learning objectives

This is what you will know or be able to do by the end of the lesson.

## Further investigation

Take your learning from the lesson further!

**Real-world examples**

Over 100 years ago, programs for the first digital computers were written using punched cards or paper tape. A hole represented 1 and no hole represented 0.

**Binary place value tables**

A place value table allows a decimal number to be converted to its binary equivalent. When converting computer data, decimal numbers are referred to as **denary** numbers. You will learn more about converting between denary and binary numbers later in this unit (see pages 88-91).

This is a 4-bit place value table, which means it can show numbers with four binary digits. The highest denary number a 4-bit binary sequence can represent is 15.

Denary number (Decimal)	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

When a computer represents 4-bit data, each number is created by turning on (1) or off (0) the binary switch that represents each of the values 8, 4, 2 and 1. Look at these two examples of equivalent binary and denary numbers:

Binary: 0011  
Denary:  $0 \times 8 + 0 \times 4 + 2 + 1 = 3$

Binary: 1011  
Denary:  $8 + 0 + 2 + 1 = 11$

**Further investigation**

- Experiment with creating your own image grids like the ones shown in this lesson.
- Investigate the meaning of the term 'denary'.

**Success criteria**

- I can explain how computers use binary numbers.
- I am able to create simple 1-bit binary graphics.
- I can use a 4-bit binary place value table.

## Success criteria

What has been understood from the lesson.

We hope you will find this book useful in developing your knowledge of digital technology, its effective use of applications and in supporting future learning.

Each topic includes easy to understand theory, real-world examples, and ideas for further investigation. You can also test your knowledge of keywords and regular exam-quality questions with supported answers. A checkpoint at the end of each lesson is a quick and easy way to check your own understanding.

### Assessment pages

Example of exam question and answer midway and end of unit to help with exam preparation.


Unit 6 Programming Part 2 Mid-unit assessment

## Unit 6 Mid-unit assessment

**Typical 4-mark exam question**

You have been asked to help build a new school records system for storing data about all the students in the school. You are going to be working with a range of data types.

Explain what is meant by the term data type and state at least two examples of data types.



**Specimen 4-mark answer**

Data types are used to identify any information stored in a program. If the data type is not correctly identified, the data cannot be processed by the computer.

Two examples of data types are whole number integers and Boolean.

142

Unit 6 Programming Part 2 End-of-unit

**What good things can we see in this answer?**

- The answer includes the key terms: joined, string, processed and variable.
- The process of concatenation is clearly described.
- An example of concatenation has been provided joining together a first and second name.

**Which parts of the answer could be better?**

- The first sentence defines concatenation as joining two strings, but two or more strings can be joined.
- The answer does not give a reason for joining the two names together.

**How can we improve this answer?**

- Improve the first paragraph by stating that two or more strings can be joined together.
- Give a reason for joining together the first and second names. For example, the full name could be displayed on screen or added to a printed report.

153

### Analysis

Evaluation of example answer at mid-unit assessment and end-of-unit assessment to hone analytical skills and provide useful guidance.

### End-of-unit checklist

Checklist at the end of every unit to quickly assess your understanding and progress.

Unit 5: Programming Part 1 End-of-unit assessment

**End-of-unit checklist**

- I know what an algorithm and a computer program are.
- I know that there are different styles of programming and many different programming languages.
- I know the purpose of text-based and visual programming languages and why pseudocode is used in planning programs.
- I know what BIDMAS is and why it is used.
- I can create simple programs in the Python programming language.
- I can create simple sorts and use arithmetic and relational operators in a simple program.
- I can search a simple database using programming terms.
- I can describe the purpose of computer models and simulations and their advantages and disadvantages.
- I can use sequences, selection and iteration in a simple Python program.
- I know what a syntax error is and can give some examples.
- I know how to find and fix syntax errors.
- I know what a subprogram is and I can use pre-existing subprograms in a simple program.

**What good things can we see in this answer?**

- The answer includes some key terms: rules, function and spelling.
- There is a clear description of what a syntax error is, and an explanation that it prevents the program from running.
- Two tips have been given, as required.

**Which parts of the answer could be better?**

- It would be helpful to include examples of spelling or character mistakes in the first paragraph.
- The second sentence is a little vague. It is not clear what 'a particular part' refers to.

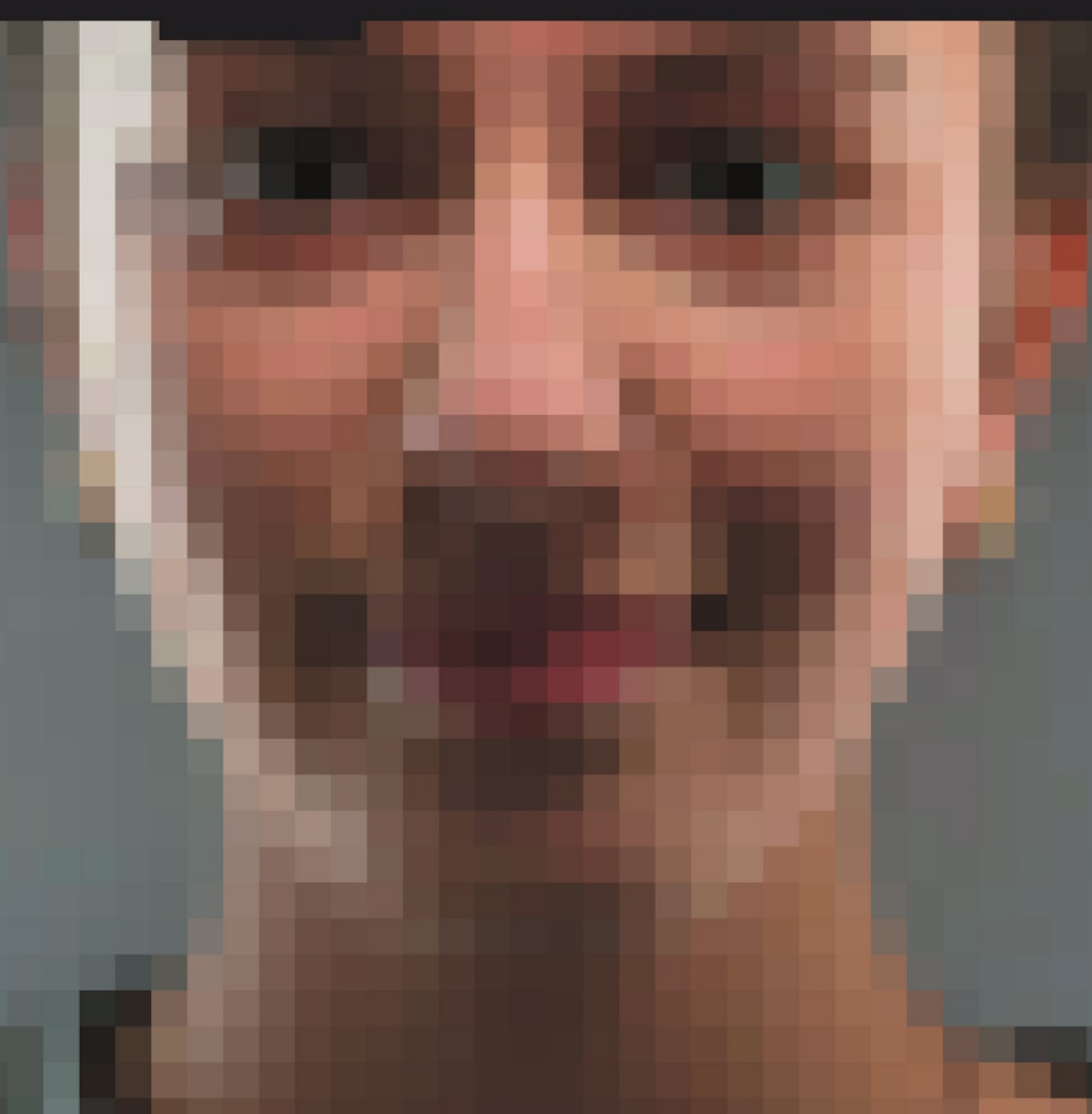
**How can we improve this answer?**

- Improve the first paragraph by giving examples of syntax errors, such as misspelling the function print or missing out a bracket.
- In the second sentence, explain that an incorrect function will halt the program at that point until the error is resolved.
- Alternative tips for finding syntax errors could include checking the case and checking the layout of functions.

129

# Unit 1

## Binary and online graphics



You are already familiar with the use of multimedia elements in your classwork, for example a presentation can contain text, images, sound, and video. But how do computers store them? You will learn about how these files are created, saved, and processed and how binary isn't just a short sequence of zeros and ones.

### **Key objectives:**

1. Learn about vector graphics and how they are created.
2. Learn about bitmap images and how they are created.
3. Learn about the properties of vector and bitmap images and the differences between them.
4. Be able to create simple vector and bitmap graphics.
5. Learn about the link between binary and pixel-based images.
6. Learn about how colour is displayed in binary.

### **By the end of the unit you will:**

- be able to create simple vector graphics
- be able to create simple bitmap images
- be able to recommend either vector or bitmap graphics for a given scenario
- convert a bitmap into binary and back again.

# Vector graphics

## Related topics

- Algorithms and programming
- Creating graphics for a target audience

## Key words

coordinates  
file size  
geometric shapes

image quality  
scale down  
scale up  
vector

## Learning objectives

- Understand the nature and purpose of a vector image.
- Understand how a vector image can be created using coordinates.
- Know how and why vectors are used for specific situations.

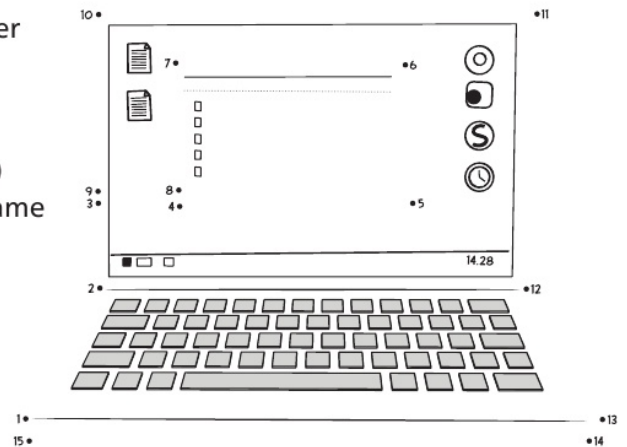
## What are vectors?

A **vector** is a type of image, created on a computer and used in graphic design all around the world. Consider a simple dot-to-dot image where the image is generated by following a series of coordinates (x and y positions from a fixed point) from start to finish. A vector image follows the same principle and requires the following instructions. These instructions form a simple algorithm.

- A starting point
- An end point
- The type of line
- Any curves or angles
- Line and fill colours.

Vector graphics have some key benefits over other types of graphic. These include:

- Ability to be scaled without losing **image quality**
- Small file size
- Easily edited
- Ideally suited for **geometric shapes**.



## Real-world examples

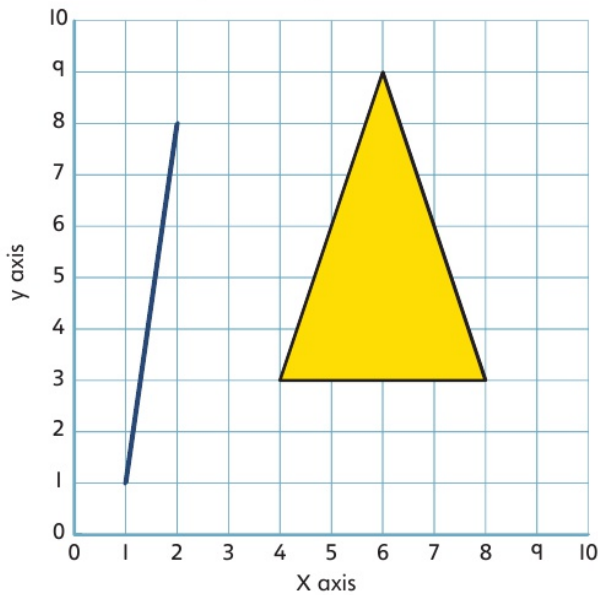
Vectors are used by graphic artists in many industries around the world. These include:

- 2D and 3D animation, from television shows to movies.
- Gaming – many of the early games were purely vector-based, however they are still used within modern console games.
- Advertising, from logos to cartoon characters.



## Creating a vector graphic

Below is an example of how to create two simple vector graphics using a series of simple instructions. By plotting coordinates and then joining the dots, a shape is made.



### Graphic 1 – Blue line

Start: X1,Y1

End: X2,Y8

Colour: Blue

### Graphic 2 – Yellow Triangle

Start: X4,Y3

Mid: X6,Y9

End: X8,Y3

Fill Colour: Yellow

## Scaling a vector graphic

One of the key benefits of a vector graphic is that it can be scaled to any size, big or small, without ever losing image quality.

### Graphic 3 – Yellow rectangle

Start: X1,Y1

End: X2,Y3

### Graphic 4 – Yellow rectangle x 2

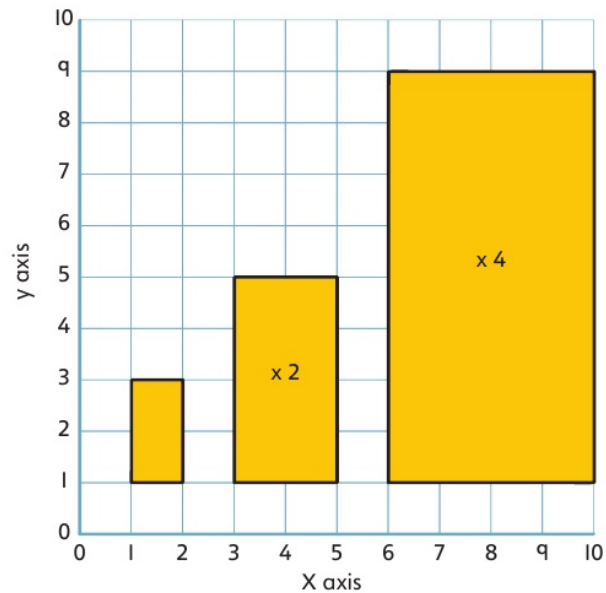
Start: X3,Y1

End: X5,Y5

### Graphic 5 – Yellow rectangle x 4

Start: X6,Y1

End: X10,Y9



## Further investigation

- Create some similar shapes of your own - **scale up** or **scale down**.
- Investigate further uses of vector graphics in industry.

## ★ Success criteria

- I know what a vector graphic is.
- I know some of the benefits of vector graphics.
- I know how a simple graphic can be created using coordinates.
- I can scale a very simple vector graphic.

# Creating a vector graphic

## Related topics

- Algorithms and programming
- Creating graphics for a target audience

## Key words

algorithm  
coordinates

image quality

scaling

vector

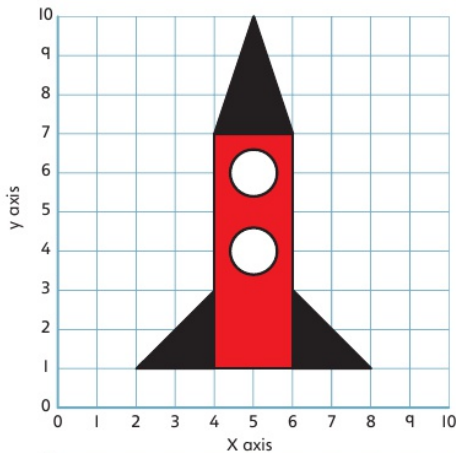
## Learning objectives

- Understand how a graphic can be created using multiple points.
- Demonstrate understanding of how to create a simple recognisable vector graphic.

## Creating graphics using an algorithm

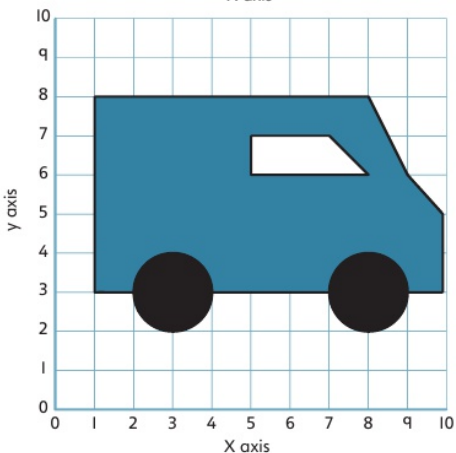
An **algorithm** is simply a set of instructions so a **vector** graphic can be plotted using a series of **coordinates** in a similar style to a dot-to-dot image.

Each line of instructions in the graphics shown creates a different shape. In the previous examples, each shape had a start and finish point. In these examples, multiple points are plotted, to create a triangle, for example. Circles are also created using the letter R to represent a radius from the XY centre.



### Graphic 1: Rocket

Black triangle left:  
X2,Y1; X4,Y3; X4,Y1  
Black triangle right:  
X6,Y1; X6,Y3; X8,Y1  
Red rectangle:  
X4,Y1; X4,Y7; X6,Y7; X6,Y1  
Black nose cone:  
X4,Y7; X5,Y10; X6,Y7  
White window 1:  
X5,Y6,R0.5  
White window 2:  
X5,Y4,R0.5



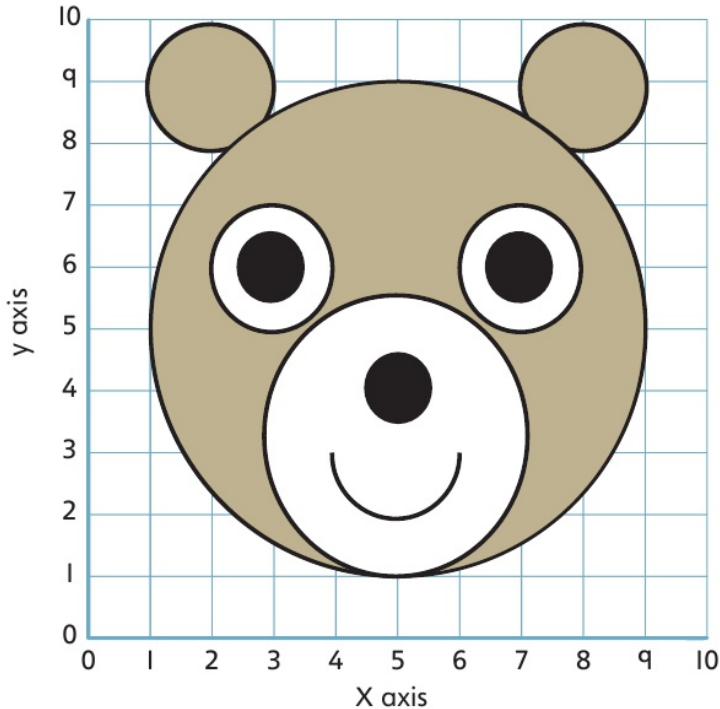
### Graphic 2: Car

Blue car body:  
X1,Y3; X1,Y8; X8,Y8  
X9,Y6; X10,Y5; X10,Y3  
White window:  
X5,Y6; X5,Y7; X7,Y7; X8,Y6  
Black wheel 1:  
X3,Y3,R1  
Black wheel 2:  
X8,Y3,R1

### Real-world examples

The Scratch visual coding app has a pen tool that can be used to create simple vector graphics using coordinates in a similar style to this page.

### Graphic 3: Animal



Brown head:  
X5,Y5,R4  
Brown left ear:  
X2,Y9,R1  
Brown right ear:  
X8,Y9,R1  
White left eye:  
X3,Y6,R1  
White right eye:  
X7,Y6,R1  
Black left eye:  
X3,Y6,R0.5  
Black right eye:  
X7,Y6,R0.5  
White snout:  
X5,Y2,R2  
Black nose:  
X5,Y4,R0.5  
Black mouth arc:  
X4,Y3; X5,Y2; X6,Y3

### Real-world examples

A graphics plotter is a device that follows a series of instructions to create a graphic, for example architectural drawings. The pen is lowered onto the paper as required and lifted when it needs to start another shape.



### Further investigation

- Investigate software packages designed to create vector graphics.
- Experiment with creating shapes of your own choice.

### Success criteria

- I know how shapes can be created using a set of instructions.
- I can create a graphic of multiple parts using a set of instructions.

# Bitmaps

## Related topics

- Digital images
- Creating graphics for a target audience

## Key words

bitmap

file size

gradient

image quality

pixel

pixels per inch (PPI)

resolution

## Learning objectives

1. Understanding of what a bitmap graphic is.
2. Understand the basic structure of a bitmap.
3. Understand the link between resolution and image quality

## What is a bitmap?

A **bitmap** is an image file made up of dots, or pixels. Each **pixel** can be a different colour. The greater the number of pixels, the more detailed the image.

These two images both show a pineapple. The one on the left is a vector; the one on the right is a bitmap.



Bitmaps can be used for:

1. Digital photographs, either taken with a digital camera or scanned using a scanner.
2. High quality original game graphics.

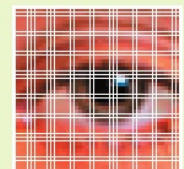
## Bitmap structure and terminology

When exploring bitmaps, it is important to understand the following key terms:

1. **Pixel:** short for picture elements. Each pixel is a tiny coloured square that makes up an image.
2. **Resolution:** relates to the number of pixels within the image. The more detailed the image, the higher the resolution is required.
3. **Gradient:** describes the change from one colour to another, fading from black to white for example with the pixels moving from black to grey to white.

## Real-world examples

Bitmaps are often referred to as raster images. Common bitmap filetypes include the following: BMP, JPG, GIF, TIFF, PNG and PICT.



## Resolution and image quality

Consider the images shown. Each pair show the same image but one is clearly sharper than the other.



The key difference between the images is the resolution – the number of pixels used to create the image. The unit of pixel resolution is **pixels per inch (PPI)**.

The image on the right of each pair has a resolution of 300 PPI and the image on the left is 10 PPI.

The **file size** of the images is related to the resolution – the higher the resolution, the larger the file size.

This means that increasing the physical size of any bitmap will reduce the resolution, lowering the **image quality**.

### 📌 Real-world examples



Games consoles are often compared based on their technical specifications. At the time of writing, the most recent Xbox and PlayStation consoles can play games at an 8K resolution of  $7680 \times 4320$  pixels. This means controlling over 33 millions pixels at once!

### 🔍 Further investigation

- Open some images using photo-editing software and experiment by changing the resolution.
- Investigate the number of pixels a smartphone or tablet can display.

### ★ Success criteria

- I know what a bitmap is.
- I know how a bitmap is made.
- I can describe the link between resolution and image quality.