**CURRICULUM OVERVIEW 2024 – 2025**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **YR 8** | **Autumn 1** | **Autumn 2** | **Spring 1** | **Spring 2** | **Summer 1** | **Summer 2** |
| **Content** | Media – Vector graphics | Computing systems | Developing for the web | Representations – from clay to silicon | Mobile app development | Introduction to Python programming |
| **Key new knowledge** | - Draw basic shapes (rectangle, ellipse, polygon, star) with different properties (fill and stroke, shape-specific attributes)- Manipulate individual objects (select, move, resize, rotate, duplicate, flip, z-order)- Manipulate groups of objects (select, group/ungroup, align, distribute)- Combine paths by applying operations (union, difference, intersection)- Convert objects to paths- Draw paths- Edit path nodes- Combine multiple tools and techniques to create a vector graphic design- Explain what vector graphics are- Provide examples where using vector graphics would be appropriate- Peer assess another pair’s project work- Improve your own project work based on feedback- Complete a summative assessment | - Recall that a general-purpose computing system is a device for executing programs- Recall that a program is a sequence of instructions that specify operations that are to be performed on data- Explain the difference between a general-purpose computing system and a purpose-built device- Describe the function of the hardware components used in computing systems- Describe how the hardware components used in computing systems work together in order to execute programs- Recall that all computing systems, regardless of form, have a similar structure (‘architecture’)- Analyse how the hardware components used in computing systems work together in order to execute programs- Define what an operating system is, and recall its role in controlling program execution- Describe the NOT, AND, and OR logical operators, and how they are used to form logical expressions- Use logic gates to construct logic circuits, and associate these with logical operators and expressions- Describe how hardware is built out of increasingly complex logic circuits- Recall that, since hardware is built out of logic circuits, data and instructions alike need to be represented using binary digits- Provide broad definitions of ‘artificial intelligence’ and ‘machine learning’- Identify examples of artificial intelligence and machine learning in the real world- Describe the steps involved in training machines to perform tasks (gathering data, training, testing)- Describe how machine learning differs from traditional programming- Associate the use of artificial intelligence with moral dilemmas- Explain the implications of sharing program code | - Describe what HTML is- Use HTML to structure static web pages- Modify HTML tags using inline styling to improve the appearance of web pages- Display images within a web page- Apply HTML tags to construct a web page structure from a provided design- Describe what CSS is- Use CSS to style static web pages- Assess the benefits of using CSS to style pages instead of in-line formatting- Describe what a search engine is- Explain how search engines ‘crawl’ through the World Wide Web and how they select and rank results- Analyse how search engines select and rank results when searches are made- Use search technologies effectively- Discuss the impact of search technologies and the issues that arise by the way they function and the way they are used- Create hyperlinks to allow users to navigate between multiple web pages- Implement navigation to complete a functioning website- Complete summative assessment | - List examples of representations- Recall that representations are used to store, communicate, and process information- Provide examples of how different representations are appropriate for different tasks- Recall that characters can be represented as sequences of symbols and list examples of character coding schemes- Measure the length of a representation as the number of symbols that it contains- Provide examples of how symbols are carried on physical media- Explain what binary digits (bits) are, in terms of familiar symbols such as digits or letters- Measure the size or length of a sequence of bits as the number of binary digits that it contains- Describe how natural numbers are represented as sequences of binary digits- Convert a decimal number to binary and vice versa- Convert between different units and multiples of representation size- Provide examples of the different ways that binary digits are physically represented in digital devices- Apply all of the skills covered in this unit | - Identify when a problem needs to be broken down- Implement and customise GUI elements to meet the needs of the user- Recognise that events can control the flow of a program- Use user input in an event-driven programming environment- Use variables in an event-driven programming environment- Develop a partially complete application to include additional functionality- Identify and fix common coding errors- Pass the value of a variable into an object- Establish user needs when completing a creative project- Apply decomposition to break down a large problem into more manageable steps- Use user input in a block-based programming language- Use a block-based programming language to create a sequence- Use variables in a block-based programming language- Use a block-based programming language to include sequencing and selection- Use user input in a block-based programming language- Use variables in a block-based programming language- Reflect and react to user feedback- Use a block-based programming language to include sequencing and selection- Use user input in a block-based programming language- Use variables in a block-based programming language- Evaluate the success of the programming project | - Describe what algorithms and programs are and how they differ- Recall that a program written in a programming language needs to be translated in order to be executed by a machine- Write simple Python programs that display messages, assign values to variables, and receive keyboard input- Locate and correct common syntax errors- Describe the semantics of assignment statements- Use simple arithmetic expressions in assignment statements to calculate values- Receive input from the keyboard and convert it to a numerical value- Use relational operators to form logical expressions- Use binary selection (if, else statements) to control the flow of program execution- Generate and use random integers- Use multi-branch selection (if, elif, else statements) to control the flow of program execution- Describe how iteration (while statements) controls the flow of program execution- Use iteration (while loops) to control the flow of program execution- Use variables as counters in iterative programs- Combine iteration and selection to control the flow of program execution- Use Boolean variables as flags |
| **Assessments** | Formative questioning andteacher observation.Summative end of unitassessment. | Formative questioning andteacher observation.Summative end of unitassessment. | Formative questioning andteacher observation.Summative end of unitassessment. | Formative questioning andteacher observation.Summative end of unitassessment. | Formative questioning andteacher observation.Summative end of unitassessment. | Formative questioning andteacher observation.Summative end of unitassessment. |