# The National Museum of Computing

## A-Level Sessions – Curriculum Links

### Tour – Valve to Chip (Computing in Context)

**What you will do:** Students will see the museum’s key exhibits: World War 2 Codebreaking machines, The Bombe, Tunny and Colossus; First Generation Computers EDSAC & WITCH, 1960s and 1970s Elliott 803, 903 and mainframes; personal computers and retro games.

**Learning Objective:** understand computing development of historical and technological context.

**Curriculum Links: Computer Science**
- Boolean Operations
- Algorithms, abstraction and automation
- Caesar, Vernam cipher
- Fundamentals of computer organisation and architecture

### Tour – Digital Revolution (Hardware, Systems, Fundamentals)

**What you will do:** Students will see: the world’s first semi programmable electronic computer; First generation computers, transistor computers, PCs and retro games. Explore how computers work and how the technology has developed.

**Learning Objective:** Student to understand how computers work.

**Curriculum Links: Computer Science**
- Assembler and ASCII
- Fundamentals of computer organisation and architecture
- Stored Program Concept
- Secondary Storage Devices

### Tour Codebreaking Machines (Cryptography)

**What you will do?** Students will find out how machines were used to crack enemy codes, in WW2 the difficulties faced by those developing them, how they overcame them, as well as seeing demonstrations of the machines in action.

**Learning Objective:** Students will understand how machines deciphered codes.

**Curriculum Links: Computer Science**
- Ciphertext, plaintext and cipher key
- Algorithms, abstraction and Automation
- The use of Boolean logic in the Lorenz Cipher

### Hands on Workshops – BBC Basic Snake Game

**What you will do:** Students create a 20-line program for a game on the 1980s BBC Micros.

**Learning Objective:** Students will learn how to create simple game using a computer language.

**Curriculum Link: Computer Science**
- Constants and variables in programming language
- Computational thinking problem solving and programming
# Hands on Workshops – BBC Basic Rocket Trajectory

**What you will do:** Students will learn how to create simple game using a computer language. This session is most suitable for students with some experience of programming using languages.

**Learning Objective:** learn how to create simple program to map rocket trajectory using BBC Basic. This session is most suitable for students with some experience of programming using languages.

**Curriculum Link: Computer Science**
- Constants and variables in programming language
- Computational thinking, problem solving and programming

---

# Hands on Workshop – BBC BASIC Assembly

**What will you do?** Look at the internal workings of a BBC micro consider how the processor relates to the machine code and assembly language. Look at how different levels of code can instruct a computer. Create a short incrementing program in BBC Basic and then write the same program in assembly to compare and contrast.

**Learning Objective:** Understand how assembler works and how it relates to machine code

**Curriculum Link: Computer Science**
- Working with constants and variables
- Binary, hexadecimal, machine code, assembler and higher-level languages
- Assembly language following and writing simple programs

---

# Hands on Workshops – Turing Test

**What you will do:** Students use a networked chat bot to discover whether they are talking to a human or a machine. In a real-life test for artificial intelligence as proposed by Alan Turing. Students create their own chat bot using artificial intelligence mark-up language (AIML) Rivescript.

**Learning Objective:** Students explore how artificial intelligence has developed and how it is used in students’ everyday life.

**Curriculum Link: Computer Science**
- Constants and variables in programming language
- Computational thinking problem solving and programming

---

# Hands on Workshop – Power of Primes

**What you will do:** Students will explore the use of prime numbers in cryptography through 2 real world examples. The Lorenz cipher and RSA cryptography. After exploring the theoretical underpinnings. Students will use a Lorenz gear visualisation to test the theory. Students will also use a special designed program to take them through the steps used in RSA encryption. They will swap messages using public and private keys.

**Learning Objective:** Students to understand cryptographic systems used to secure data.

**Curriculum Link: Computer Science**
- Asymmetric cryptography
- Ciphertext, plaintext and cipher key
- Caesar and Vernam cipher
- Random number generation
The National Museum of Computing  
A-Level Sessions – Curriculum Links

<table>
<thead>
<tr>
<th>Hands on Workshop - Beating Impossible Odds</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What you will do</strong> – look at how an Enigma machine works looking at how the machine builds complexity. Working in teams of ‘human enigmas’ you will send and receive messages.</td>
</tr>
<tr>
<td><strong>Learning Objective:</strong> understand how the logic circuits of Enigma machine create a cryptography.</td>
</tr>
</tbody>
</table>
| **Curriculum Link: Maths**  
  • Sequences, series, arithmetic progressions  
  • Iteration |

<table>
<thead>
<tr>
<th>Hands on Workshop – Statistical Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What you will do</strong> – Students will discover how the Lorenz machine creates complexity and how Bill Tutte used statistics and probability to hack the cipher through pattern analysis.</td>
</tr>
<tr>
<td><strong>Learning Objective:</strong> Students to understand how the Lorenz cipher creates complexity on a massive scale and how Bill Tutte hacked the cipher to create a diagram and an algorithm</td>
</tr>
</tbody>
</table>
| **Curriculum Link: Computer Science and Maths**  
  • What is meant by encryption, cipher, plain text and cipher text  
  • Solving logic problems through computational thinking  
  • Pattern Recognition |