**Chapter 1: Notational Systems**

1. **Binary -**The binary numbering system is a way of counting using only two digits: **0** and **1**. It’s the language computers use to store and process information because it’s easy for electronics to understand—on (1) and off (0). Each digit in binary is called a **bit**, and by combining bits, we can represent numbers, letters, and even pictures! For example, the number **5** in binary is written as **101**.
2. **The hexadecimal**-numbering system (or "hex") is a way of counting that uses **16 symbols**: the digits **0 to 9**, and the letters **A to F**. Each letter stands for numbers 10 to 15. Hex is often used in computers because it’s shorter and easier to read than binary. For example, the number **10** in decimal is **A** in hex, and the number **255** in decimal is **FF** in hex.
3. **Decimal** - The decimal numbering system is the way we usually count in everyday life. It uses **10 digits**: **0, 1, 2, 3, 4, 5, 6, 7, 8, and 9**. Each digit’s place in a number represents a power of **10** (like ones, tens, hundreds, etc.). For example, in the number **345**, the **3** is in the hundreds place, the **4** is in the tens place, and the **5** is in the ones place.

**Videos:**

* [**Binary Numbers and Base Systems as Fast as Possible**](https://youtu.be/LpuPe81bc2w?si=4Og54PVWiENRA4x8)
* **[Hexadecimal number system | Applying mathematical reasoning | Pre-Algebra | Khan Academy](https://youtu.be/4EJay-6Bioo?si=A6a4TVthRfdzCsDR)**
* [**Introduction to number systems and binary | Pre-Algebra | Khan Academy**](https://youtu.be/ku4KOFQ-bB4?si=5AlXJV5ZGYx8jaZO)
* [**The binary number system**](https://youtu.be/sXxwr66Y79Y?si=n0zabgdB-LvA7Zpb)
* [**Binary, Decimal and Hexadecimal Number Systems**](https://youtu.be/_97OwCkjh3c?si=Ze6adwSiTqtDHswD)

**Other References**

* [Numbers](https://www.csfieldguide.org.nz/en/chapters/data-representation/numbers/)
* [Binary Number System](https://www.mathsisfun.com/binary-number-system.html)
* [GeekForce – Binary](https://www.geeksforgeeks.org/binary-number-system/)

**Turning Decimal Hexadecimal Conversion**

To convert the decimal number 3284 into hexadecimal, follow these steps:

1. **Divide the decimal number by 16 and keep track of the quotient and the remainder.**
2. **Continue dividing the quotient by 16 until the quotient is 0, recording each remainder.**
3. **The hexadecimal number is the sequence of remainders read from bottom to top.**

Let's perform the conversion:

1. 3284÷16=205 remainder 4
2. 205÷16 = 12 remainder 13 (which is D in hexadecimal)
3. 12÷16=0 remainder 12 (which is C in hexadecimal)

Reading the remainder from bottom to top, we get C D 4.

**Example 2**

To convert the decimal number 3282 into hexadecimal, follow these steps:

Let's perform the conversion:

1. 3282÷16=205 remainder 2
2. 205÷16=12 remainder 13 (which is D in hexadecimal)
3. 12÷16=0 remainder 12 (which is C in hexadecimal)

Reading the remainders from bottom to top, we get C D 2.

**Example 3**

To convert the decimal number 67 into hexadecimal, follow these steps:

Let's perform the conversion:

1. 67÷16=467 ÷ 16 = 467÷16=4 remainder 3
2. 4÷16=04 ÷ 16 = 04÷16=0 remainder 4

Reading the remainder from bottom to top, we get 43

**Practice on each of the numbers:**

**Practice 1**: To convert the decimal number **269** into hexadecimal, follow these steps:

1. Divide the decimal number by 16 and keep track of the quotient and the remainder.
2. Continue dividing the quotient by 16 until the quotient is 0, recording each remainder.
3. The hexadecimal number is the sequence of remainders read from bottom to top.

Let's perform the conversion:

1. 269÷16= 16, remainder 13 (which is D in hexadecimal)
2. 16÷16=1 remainder 0
3. 1÷16=0 remainder 1

Reading the remainder from bottom to top, we get 10D

\*\*\*

**Practice** **2**: To convert the decimal number **7183** into hexadecimal, follow these steps:

1. Divide the decimal number by 16 and keep track of the quotient and the remainder.
2. Continue dividing the quotient by 16 until the quotient is 0, recording each remainder.
3. The hexadecimal number is the sequence of remainders read from bottom to top.

Let's perform the conversion:

1. 7183÷16 = 448 remainder 15 (which is F in hexadecimal)
2. 448÷16=28 remainder 0
3. 28÷16=1 remainder 12 (which is C in hexadecimal)
4. 1÷16=0 remainder 1

Reading the remainder from bottom to top, we get 1C0F.

\*\*\*

**Practice** **3:** To convert the decimal number **3985** into hexadecimal, follow these steps:

1. Divide the decimal number by 16 and keep track of the quotient and the remainder.
2. Continue dividing the quotient by 16 until the quotient is 0, recording each remainder.
3. The hexadecimal number is the sequence of remainders read from bottom to top.

Let's perform the conversion:

1. 3985÷16=449 remainder 1
2. 249÷16=15 remainder 9
3. 15÷16=0 remainder 15 (which is F in hexadecimal)

Reading the remainder from bottom to top, we get F91.

\*

**Practice** **4:** To convert the decimal number **1639** into hexadecimal, follow these steps:

1. Divide the decimal number by 16 and keep track of the quotient and the remainder.
2. Continue dividing the quotient by 16 until the quotient is 0, recording each remainder.
3. The hexadecimal number is the sequence of remainders read from bottom to top.

Let's perform the conversion:

1. 1639÷16=102 remainder 7
2. 102÷16=6 remainder 6
3. 6÷16=0 remainder 6

Reading the remainder from bottom to top, we get 667

\*\*\*

**Practice 5:** To convert the decimal number **921** into hexadecimal, follow these steps:

Let's perform the conversion:

1. 921÷16 = 57 remainder 9
2. 57÷16=3, remainder 9
3. 3÷16 = 0, remainder 3

Reading the remainder from bottom to top, we get 399.

**Practice 6:** To convert the decimal number **182** into hexadecimal, follow these steps:

Let's perform the conversion:

1. 182÷16=remainder 6
2. 11÷16= (which is B in hexadecimal)

Reading the remainder from bottom to top, we get B6.

So, the decimal number 182 is B6 in hexadecimal.

### **Hex Data**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DEC** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| **HEX** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

### **Binary**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **=** | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | **=** | 255 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | **=** | 138 |
|  |  |  |  |  |  |  |  |  | 8 |
|  |  |  |  |  |  |  |  |  | 171 |
|  |  |  |  |  |  |  |  |  | 88 |
|  |  |  |  |  |  |  |  |  | 67 |

### **When to use the Hexadecimal number system**

### **Color Codes in Web Design:** When creating websites, colors are often defined using hexadecimal codes in Cascading Style Sheets (CSS). Each color is represented as a combination of red, green, and blue (RGB) values, where the intensity of each component is specified using a **hexadecimal value**.

For instance:

* **#FF5733** represents a shade of orange.
  + **FF** (hexadecimal for 255) = maximum red intensity.
  + **57** (hexadecimal for 87) = medium green intensity.
  + **33** (hexadecimal for 51) = low blue intensity.

Hexadecimal is ideal here because it’s compact and maps perfectly to the 8-bit binary values that computers use internally. Each pair of hex digits (e.g., "FF") corresponds to 8 bits, making it efficient to represent and manipulate colors in digital systems.

This use of hexadecimal is widespread in web design, digital imaging, and video game development, as it provides a precise and human-readable way to define millions of colors.

### When to use a **binary number system**

### **Computer Memory and Data Representation:** Computers use binary to store and process all types of data, including text, images, videos, and instructions. This is because binary corresponds directly to how computer hardware operates—using electrical signals to represent **on (1)** and **off (0)** states.

For example:

* **Text Representation**:  
  Each letter or character you type is converted into binary using a system like **ASCII**.
  + The letter **A** is represented as **01000001** in binary (corresponding to its ASCII code, 65 in decimal).
  + The letter **B** is **01000010** (ASCII code 66).
* **Images**:  
  Every pixel in an image is represented in binary. For example, a black-and-white image might use **1** for a white pixel and **0** for a black pixel. A more complex image uses binary to store RGB values for each pixel.
* **Storage**:
* Binary is used to measure storage in devices. For instance:
  + A kilobyte (KB) = **1024 bytes**, and each byte is made up of **8 bits**, which are binary digits (0s and 1s).
  + So, a 1 KB file uses **8192 binary digits** to store its information.

**Turning Hexadecimal to Decimal**

Turning a hexadecimal number into a decimal number involves multiplying each digit of the hexadecimal number by 16 raised to the power of its position (counting from right to left, starting at 0). Here's a step-by-step guide:

1. Write down the hexadecimal number.
2. Starting from the right, assign positions to each digit, starting at 0.
3. Convert each hex digit to its decimal equivalent (0-9 and A-F where A=10, B=11, ..., F=15).
4. Multiply each decimal equivalent by 16 raised to the power of its position.
5. Add all these values together to get the decimal number.

Let's convert the hexadecimal number B6 to decimal as an example:

1. B6 in hex is B×16 + 6 times 16
2. B is 11 in decimal, and 6 is 6 in decimal.
3. Calculate each part:
   * B×16 =11×16 =176
   * 6×16 = 6×1=6
4. Add these values together: 176+6=182, So, the hexadecimal number B6 is 182 in decimal.

**What is the Hex Equivalent?**

|  |  |
| --- | --- |
| **Decimal** | **Hex** |
| 35 |  |
| 67 |  |
| 17 |  |
| 90 |  |
| 5 |  |
| 8 |  |
| 118 |  |
| 63 |  |
| 240 |  |
| 160 |  |

Dr. Thomas Russell

719-213-4610

Email: thruss09@gmail.com

Website: semtechedu.org