

# 1.8 ASCII code and Cast operator

In my video, I will show

## 1. Encoding a Simple message to a hidden message

1.1 Simple message

1.2 Cast operator (char) and (double)

1.3 ASCII codes of your simple message (matB)

## 2. Encoding

- Define encryption matrix matA
- $\text{matC} = \text{matA} * \text{matB}$

## 3. Decode your friend's message

- matA
- A decoded message of your friend's

## 4. try-catch

# ASCII code

## American Standard Code for Information Interchange

| Char | Dec | Oct  | Hex  | Char | Dec | Oct  | Hex  | Char | Dec | Oct  | Hex  |
|------|-----|------|------|------|-----|------|------|------|-----|------|------|
| (sp) | 32  | 0040 | 0x20 | @    | 64  | 0100 | 0x40 | `    | 96  | 0140 | 0x60 |
| !    | 33  | 0041 | 0x21 | A    | 65  | 0101 | 0x41 | a    | 97  | 0141 | 0x61 |
| "    | 34  | 0042 | 0x22 | B    | 66  | 0102 | 0x42 | b    | 98  | 0142 | 0x62 |
| #    | 35  | 0043 | 0x23 | C    | 67  | 0103 | 0x43 | c    | 99  | 0143 | 0x63 |
| \$   | 36  | 0044 | 0x24 | D    | 68  | 0104 | 0x44 | d    | 100 | 0144 | 0x64 |
| %    | 37  | 0045 | 0x25 | E    | 69  | 0105 | 0x45 | e    | 101 | 0145 | 0x65 |
| &    | 38  | 0046 | 0x26 | F    | 70  | 0106 | 0x46 | f    | 102 | 0146 | 0x66 |
| '    | 39  | 0047 | 0x27 | G    | 71  | 0107 | 0x47 | g    | 103 | 0147 | 0x67 |
| (    | 40  | 0050 | 0x28 | H    | 72  | 0110 | 0x48 | h    | 104 | 0150 | 0x68 |
| )    | 41  | 0051 | 0x29 | I    | 73  | 0111 | 0x49 | i    | 105 | 0151 | 0x69 |
| *    | 42  | 0052 | 0x2a | J    | 74  | 0112 | 0x4a | j    | 106 | 0152 | 0x6a |
| +    | 43  | 0053 | 0x2b | K    | 75  | 0113 | 0x4b | k    | 107 | 0153 | 0x6b |
| ,    | 44  | 0054 | 0x2c | L    | 76  | 0114 | 0x4c | l    | 108 | 0154 | 0x6c |
| -    | 45  | 0055 | 0x2d | M    | 77  | 0115 | 0x4d | m    | 109 | 0155 | 0x6d |
| .    | 46  | 0056 | 0x2e | N    | 78  | 0116 | 0x4e | n    | 110 | 0156 | 0x6e |
| /    | 47  | 0057 | 0x2f | O    | 79  | 0117 | 0x4f | o    | 111 | 0157 | 0x6f |
| 0    | 48  | 0060 | 0x30 | P    | 80  | 0120 | 0x50 | p    | 112 | 0160 | 0x70 |
| 1    | 49  | 0061 | 0x31 | Q    | 81  | 0121 | 0x51 | q    | 113 | 0161 | 0x71 |
| 2    | 50  | 0062 | 0x32 | R    | 82  | 0122 | 0x52 | r    | 114 | 0162 | 0x72 |
| 3    | 51  | 0063 | 0x33 | S    | 83  | 0123 | 0x53 | s    | 115 | 0163 | 0x73 |
| 4    | 52  | 0064 | 0x34 | T    | 84  | 0124 | 0x54 | t    | 116 | 0164 | 0x74 |
| 5    | 53  | 0065 | 0x35 | U    | 85  | 0125 | 0x55 | u    | 117 | 0165 | 0x75 |
| 6    | 54  | 0066 | 0x36 | V    | 86  | 0126 | 0x56 | v    | 118 | 0166 | 0x76 |
| 7    | 55  | 0067 | 0x37 | W    | 87  | 0127 | 0x57 | w    | 119 | 0167 | 0x77 |
| 8    | 56  | 0070 | 0x38 | X    | 88  | 0130 | 0x58 | x    | 120 | 0170 | 0x78 |
| 9    | 57  | 0071 | 0x39 | Y    | 89  | 0131 | 0x59 | y    | 121 | 0171 | 0x79 |
| :    | 58  | 0072 | 0x3a | Z    | 90  | 0132 | 0x5a | z    | 122 | 0172 | 0x7a |
| ;    | 59  | 0073 | 0x3b | [    | 91  | 0133 | 0x5b | {    | 123 | 0173 | 0x7b |
| <    | 60  | 0074 | 0x3c | \    | 92  | 0134 | 0x5c |      | 124 | 0174 | 0x7c |
| =    | 61  | 0075 | 0x3d | ]    | 93  | 0135 | 0x5d | }    | 125 | 0175 | 0x7d |
| >    | 62  | 0076 | 0x3e | ^    | 94  | 0136 | 0x5e | ~    | 126 | 0176 | 0x7e |
| ?    | 63  | 0077 | 0x3f | _    | 95  | 0137 | 0x5f |      |     |      |      |

1. Convert a simple message to ASCII codes

- `char[][] message = {{'J','e','s','u'},{'s','l','o','v'},{'e','s','m','e'}};`
- `matB[i][j]=(double)(message[i][j]);`

2. Print the ASCII codes

- `System.out.print((char)matB[i][j]);`

3. Define an encryption matrix matA

- `double[][] matA = {{4,3,3},{1,2,1},{1,3,4}} ;`

4. `matC= matA * matB`

5. email matA and matC to your friend.

6. Decode matC back to a simple code using matA

# (Optional) Fourier Transform