Lab 3 - Task 4: Padding Behavior Across AES Modes

Objective

Show, with a 15-byte input, that AES-ECB and AES-CBC use padding (PKCS#7) while AES-CFB and AES-OFB do not. Verify by ciphertext sizes and by successful decryption back to the original.

Setup

Environment: Ubuntu, OpenSSL available in PATH. Test input length: 15 bytes (not a multiple of AES block size 16).

```
KEY=00112233445566778899aabbccddeeff   # 16-byte (128-bit) key in hex
IV=0102030405060708090a0b0c0d0e0f10   # 16-byte IV for CBC/CFB/OFB
printf "123456789abcdef" > p15.txt   # 15 bytes
wc -c p15.txt                         # -> 15
```

Commands: Encrypt in Four Modes

```
# ECB (expects padding)
openssl enc -aes-128-ecb -e -in p15.txt -out p15_ecb.bin -K $KEY -nosalt

# CBC (expects padding)
openssl enc -aes-128-cbc -e -in p15.txt -out p15_cbc.bin -K $KEY -iv $IV -nosalt

# CFB (no padding)
openssl enc -aes-128-cfb -e -in p15.txt -out p15_cfb.bin -K $KEY -iv $IV -nosalt

# OFB (no padding)
openssl enc -aes-128-ofb -e -in p15.txt -out p15_ofb.bin -K $KEY -iv $IV -nosalt
```

Evidence: Ciphertext Sizes

```
stat -c%s p15_ecb.bin
stat -c%s p15_cbc.bin
stat -c%s p15_cfb.bin
stat -c%s p15_ofb.bin
```

Expected:
- ECB/CBC -> 16 bytes (PKCS#7 padded to full block)
- CFB/OFB -> 15 bytes (no padding; output length equals input length)

Decrypt and Verify

```
openssl enc -aes-128-ecb -d -in p15_ecb.bin -out ecb.dec -K $KEY -nosalt
openssl enc -aes-128-cbc -d -in p15_cbc.bin -out cbc.dec -K $KEY -iv $IV -nosalt
openssl enc -aes-128-cfb -d -in p15_cfb.bin -out cfb.dec -K $KEY -iv $IV -nosalt
openssl enc -aes-128-ofb -d -in p15_ofb.bin -out ofb.dec -K $KEY -iv $IV -nosalt

diff -s p15.txt ecb.dec
diff -s p15.txt cbc.dec
diff -s p15.txt cfb.dec
```

Lab 3 - Task 4: Padding Behavior Across AES Modes

```
diff -s p15.txt ofb.dec
```

All four diffs should report the files are identical (decryption correct).

Optional: Why padding is needed for ECB/CBC

```
openssl enc -aes-128-ecb -e -in p15.txt -out bad.bin -K $KEY -nosalt -nopad
# Expected: error (input not multiple of block length).
```

Observations (to include in write-up)

- AES uses a 16-byte block size. Because 15 is not a multiple of 16, block modes (ECB, CBC) need PKCS#7 padding; stream-like modes (CFB, OFB) do not use padding.
- Ciphertext length confirms this: ECB/CBC grew to 16; CFB/OFB stayed 15.
- Successful decryption in all four modes verifies correctness.

Conclusion

For non-block-multiple inputs, AES-ECB and AES-CBC add PKCS#7 padding, while AES-CFB and AES-OFB do not. This is observable via ciphertext lengths and verified by decryption back to the original message.