# Assignment 5

## SHA-1 encryption

```
import hashlib
def generate_sha1_hash(message):
    sha1 = hashlib.sha1()
    # Update the hash object with the bytes-Like ob.
    sha1.update(message.encode('utf-8'))
    # Get the hexadecimal representation of the hasi
    sha1_hash = sha1.hexdigest()
    return sha1_hash
message = "Kaushal Bharambe"
sha1_hash = generate_sha1_hash(message)
print("Original Message:", message)
print ("SHA-1 Hash Value:", sha1_hash)
```

Output:

```
Original Message: Kaushal Bharambe
SHA-1 Hash Value: c4596a71f98f58ebbe9c76346e255dde55d3d038
>
```

## Conclusion:-

SHA-1, once widely used, is now considered weak due to vulnerabilities allowing collision attacks. Stronger hash functions like SHA-256 or SHA-3 are recommended for enhanced security. Understanding SHA-1 principles is crucial for historical context and cryptographic hash functions.

# Assignment 5

## SHA-1 encryption

```
import hashlib
def generate_sha1_hash(message):
    sha1 = hashlib.sha1()
    # Update the hash object with the bytes-Like ob.
    sha1.update(message.encode('utf-8'))
    # Get the hexadecimal representation of the hasi
    sha1_hash = sha1.hexdigest()
    return sha1_hash
message = "Anushri Bhoyar "
sha1_hash = generate_sha1_hash(message)
print("Original Message:", message)
print ("SHA-1 Hash Value:", sha1_hash)
```

Output:

```
Original Message: Anushri Bhoyar
SHA-1 Hash Value: 35ae8953e3cb3f7c4f8b9efd78827c0e2f59410d
>
```

## Conclusion:-

SHA-1, once widely used, is now considered weak due to vulnerabilities allowing collision attacks. Stronger hash functions like SHA-256 or SHA-3 are recommended for enhanced security. Understanding SHA-1 principles is crucial for historical context and cryptographic hash functions.

# Assignment 5

## SHA-1 encryption

```
import hashlib
def generate_sha1_hash(message):
    sha1 = hashlib.sha1()
    # Update the hash object with the bytes-Like ob.
    sha1.update(message.encode('utf-8'))
    # Get the hexadecimal representation of the hasi
    sha1_hash = sha1.hexdigest()
    return sha1_hash
message = "Pranav Mahajan"
sha1_hash = generate_sha1_hash(message)
print("Original Message:", message)
print ("SHA-1 Hash Value:", sha1_hash)
```

Output:

```
Original Message: Pranav Mahajan
SHA-1 Hash Value: eb13b168354f1a0e83a390ebd46326c8408bc00a
>
```

## Conclusion:-

SHA-1, once widely used, is now considered weak due to vulnerabilities allowing collision attacks. Stronger hash functions like SHA-256 or SHA-3 are recommended for enhanced security. Understanding SHA-1 principles is crucial for historical context and cryptographic hash functions.