```
import hashlib
# Function to generate hash using MD5
def generate_md5_hash(message):
    md5_hash = hashlib.md5()
    md5_hash.update(message.encode('utf-8'))  # Encoding message to bytes
    return md5_hash.hexdigest()

# Function to generate hash using SHA-1
def generate_sha1_hash(message):
    sha1_hash = hashlib.sha1()
    sha1_hash.update(message.encode('utf-8'))  # Encoding message to bytes
    return sha1_hash.hexdigest()

# Function to verify message integrity
def verify_message_integrity(original_message, provided_hash, hash_algorithm='md5'):
    if hash_algorithm == 'md5':
        calculated_hash = generate_md5_hash(original_message)
    elif hash_algorithm == 'sha1':
        calculated_hash = generate_sha1_hash(original_message)
    else:
        raise ValueError("Invalid hash algorithm")
    # Compare the provided hash with the calculated hash
    if calculated_hash == provided_hash:
        return True
    else:
        return False

# Example usage
if __name__ == "__main__":
    message = "LIFE IS ALL ABOUT STRUGGLE"

    # Generate MD5 hash of the message
    md5_hash = generate_md5_hash(message)
    print(f"MD5 Hash: {md5_hash}")

    # Generate SHA-1 hash of the message
    sha1_hash = generate_sha1_hash(message)
    print(f"SHA-1 Hash: {sha1_hash}")

    # Simulate message verification
    print("\nVerifying Message Integrity (MD5):")
    is_valid = verify_message_integrity(message, md5_hash, 'md5')
    print("Message Integrity Verified:", is_valid)

    print("\nVerifying Message Integrity (SHA-1):")
    is_valid = verify_message_integrity(message, sha1_hash, 'sha1')
    print("Message Integrity Verified:", is_valid)
```

```
MD5 Hash: 6c41ec223101ed3dea43e0322e1556b2
SHA-1 Hash: 2a400eb981066c85f25f9336fad9ab71fe850853

Verifying Message Integrity (MD5):
Message Integrity Verified: True

Verifying Message Integrity (SHA-1):
Message Integrity Verified: True
```