AI Assisted Coding Ass-5.5

Lab 5: Ethical Foundations – Responsible AI Coding Practices

Name:- MD.Abdul Layeeq

HT.No:- 2303A52217

Batch:- 45

Task 1: Transparency in Algorithm Optimization (Prime Number Check)

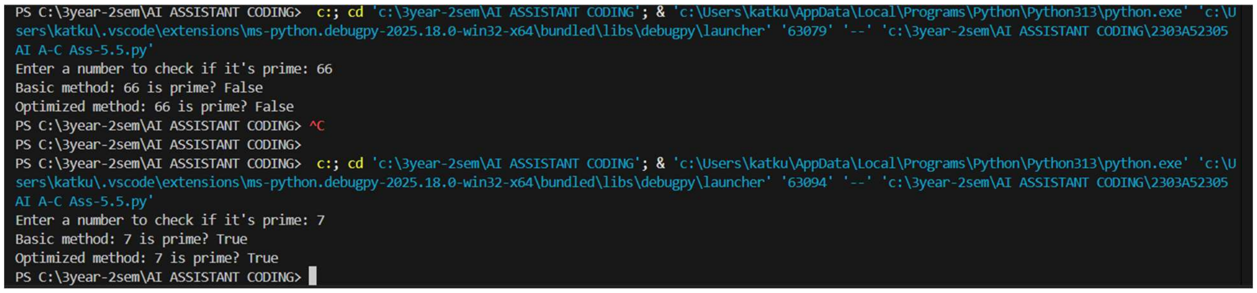
Prompt

“Generate Python code for two prime-checking methods and explain how the optimized version improves performance.”

Code:-

```
import math
def is_prime_basic(n):
  """Check if a number is prime using basic method."""
  if n <= 1:
    return False
  for i in range(2, n):
    if n % i == 0:
      return False
  return True
def is_prime_optimized(n):
  """Check if a number is prime using optimized method."""
  if n <= 1:
    return False
  if n <= 3:
    return True
    if n % 2 == 0 or n % 3 == 0:
        return False
    for i in range(5, int(math.sqrt(n)) + 1, 6):
        if n % i == 0 or n % (i + 2) == 0:
            return False
    return True
n = int(input("Enter a number to check if it's prime: "))
print(f"Basic method: {n} is prime? {is_prime_basic(n)}")
print(f"Optimized method: {n} is prime? {is_prime_optimized(n)}")
```

Output:-



```
PS C:\3year-2sem\AI ASSISTANT CODING>  c:; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\U
sers\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '63079' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-5.5.py'
Enter a number to check if it's prime: 66
Basic method: 66 is prime? False
Optimized method: 66 is prime? False
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING>  c:; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\U
sers\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '63094' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-5.5.py'
Enter a number to check if it's prime: 7
Basic method: 7 is prime? True
Optimized method: 7 is prime? True
PS C:\3year-2sem\AI ASSISTANT CODING>
```

Efficiency

Method	**Time Complexity**
Naive	**O(n)**
Optimized	**O(√n)**

Observation

The optimized method reduces the number of checks using mathematical logic (factors occur in pairs). Transparency helps understand **why** optimization works instead of blindly using faster code.

Explanation of optimization:

->The basic method checks for factors from 2 to n-1, which can be very slow for large n.

->The optimized method reduces the number of checks significantly:

1. It first eliminates numbers less than or equal to 3 and even numbers greater than 2.

2. It only checks for factors up to the square root of n, since if n is divisible by any number greater than its square root,

->The corresponding factor must be less than the square root.

3. It skips even numbers and checks only numbers of the form 6k ± 1, as all primes greater than 3 can be expressed in this form.

Task 2: Transparency in Recursive Algorithms (Fibonacci)

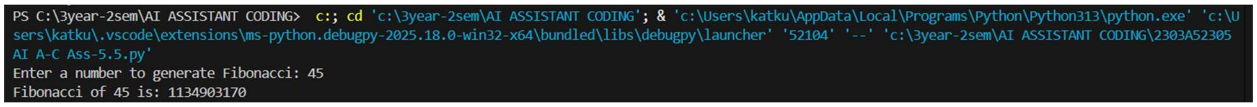
Prompt

Generate recursive Fibonacci function with explanation.

Code:-

```
def fibonacci_recursive(n):
    """Generate the nth Fibonacci number using recursion."""
    if n <= 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci_recursive(n - 1) + fibonacci_recursive(n - 2)
n = int(input("Enter a number to generate Fibonacci: "))
print(f"Fibonacci of {n} is: {fibonacci_recursive(n)}")
```

Output:-



```
PS C:\3year-2sem\AI ASSISTANT CODING>  c:; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\U
sers\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '52104' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-5.5.py'
Enter a number to generate Fibonacci: 45
Fibonacci of 45 is: 1134903170
```

Explanation

- **Base Cases:** Stop recursion at 0 and 1
- **Recursive Step:** Each call splits into two smaller subproblems
- Demonstrates how recursion builds the solution from smaller parts.**Observation**

Recursive transparency ensures students understand stack calls, not just memorized formulas.

Explanation of recursive Fibonacci:

-> The recursive approach directly implements the mathematical definition of the Fibonacci sequence.

-> It calculates F(n) by recursively calling itself for F(n-1) and F(n-2).

-> While this is intuitive and easy to understand, it is inefficient for large n because it recalculates the same values multiple times.

-> For example, to compute F(5), it computes F(4) and F(3), but F(4) itself computes F(3) again, leading to redundant calculations.

Task 3: Transparency in Error Handling

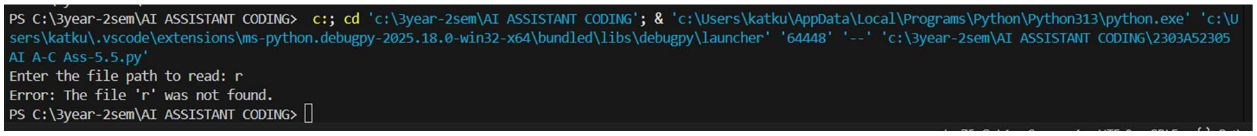
Prompt

Generate file-processing program with error handling.

Code:-

```
def read_file(file_path):
  """Read and print the contents of a file with error handling."""
  try:
    with open(file_path, 'r') as file:
      content = file.read()
      print("File Content:")
      print(content)
    except FileNotFoundError:
        print(f"Error: The file '{file_path}' was not found.")
    except IOError:
        print(f"Error: An I/O error occurred while trying to read the file '{file_path}'.")
file_path = input("Enter the file path to read: ")
read_file(file_path)
```

Output:-



```
PS C:\3year-2sem\AI ASSISTANT CODING>  c:; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\U
sers\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '64448' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-5.5.py'
Enter the file path to read: r
Error: The file 'r' was not found.
PS C:\3year-2sem\AI ASSISTANT CODING> 
```

Observation

AI often generates code without proper exceptions. Ethical coding requires handling real-world failures clearly.

Task 4: Security in User Authentication

Prompt

Generate a Python-based login system.

->Then analyze the security of the generated code by checking for:

->Plain-text password storage

->Weak validation

->Provide an improved version using:

->Password hashing

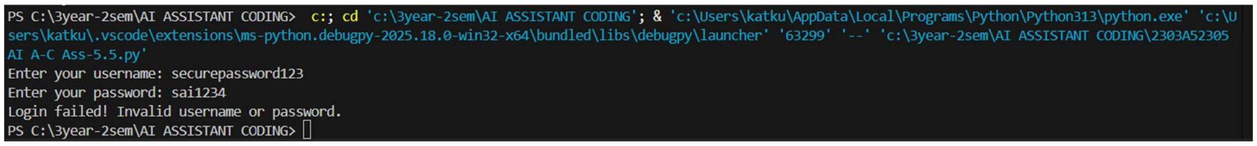
->Proper input validation

->Also include a short note on secure authentication best practices.

Code

```
import hashlib

def hash_password(password):
  """Hash a password for secure storage."""
  return hashlib.sha256(password.encode()).hexdigest()

def validate_username(username):
  """Validate username for proper format."""
  if len(username) < 5 or len(username) > 20:
    return False
  if not username.isalnum():
    return False
  return True

def login_system():
  """A simple login system with hashed passwords and input validation."""
  users = {
    "user1": hash_password("securepassword123"),
    "user2": hash_password("anothersecurepassword456")
  }
  username = input("Enter your username: ")
  password = input("Enter your password: ")
  if not validate_username(username):
    print("Invalid username format. Must be 5-20 alphanumeric characters.")
    return
  hashed_input_password = hash_password(password)
  if username in users and users[username] == hashed_input_password:
    print("Login successful!")
    else:
        print("Login failed! Invalid username or password.")
login_system()
```

Output:-



```
PS C:\3year-2sem\AI ASSISTANT CODING>  c:; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\U
sers\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '63299' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-5.5.py'
Enter your username: securepassword123
Enter your password: sai1234
Login failed! Invalid username or password.
PS C:\3year-2sem\AI ASSISTANT CODING> 
```

Flaws:

• Plain-text password
• No hashing
• Weak validation

Secure Version

```
import hashlib
def hash_password(password):
    return hashlib.sha256(password.encode()).hexdigest()
stored_user = "admin"
stored_pass_hash = hash_password("1234")
username = input("Username: ")
password = input("Password: ")
if username == stored_user and hash_password(password) == stored_pass_hash:
    print("Login successful")
else:
    print("Invalid credentials")
```

Best Practices

- Never store plain passwords
- Use hashing + salting
- Validate inputs

- Limit login attempts

Task 5: Privacy in Data Logging

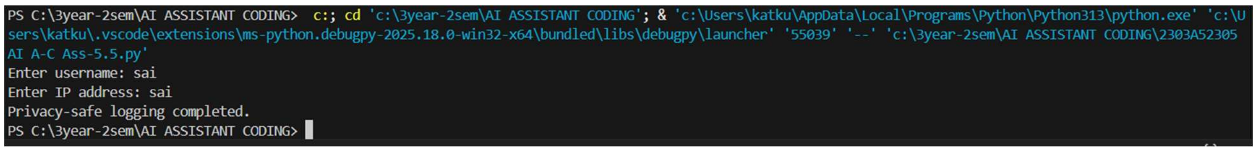
Prompt:-

Generate a Python script that logs user activity including username, IP address, and timestamp.Then analyze the privacy risks by checking whether sensitive data is logged insecurely or unnecessarily.Provide an improved version that uses minimal, anonymized, or masked logging.Also explain the principles of privacy-aware logging.

Code:-

```
import datetime

import hashlib

def anonymize_username(username):

   return hashlib.sha256(username.encode()).hexdigest()[:8]

def mask_ip(ip):

   parts = ip.split(".")

   return ".".join(parts[:2]) + ".xxx.xxx"

username = input("Enter username: ")

ip_address = input("Enter IP address: ")

safe_user = anonymize_username(username)

safe_ip = mask_ip(ip_address)

timestamp = datetime.datetime.now()

with open("log.txt", "a") as file:

   file.write(f"{safe_user}, {safe_ip}, {timestamp}\n")

print("Privacy-safe logging completed.")
```

Output:-



```
PS C:\3year-2sem\AI ASSISTANT CODING>  c:; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\U
sers\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '55039' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-5.5.py'
Enter username: sai
Enter IP address: sai
Privacy-safe logging completed.
PS C:\3year-2sem\AI ASSISTANT CODING> 
```

Principles

- Log only required data
- Mask identities
- Avoid storing raw IPs
- Follow data protection laws

Final Ethical Reflection

Area	Risk	Responsible Practice
Optimization	Blind use of AI code	Explain logic
Recursion	Misunderstanding execution	Add comments
Error Handling	Crashes	Handle exceptions
Security	Password leaks	Hashing
Privacy	Data exposure	Masking

Conclusion

AI-generated code must be **verified, secured, and explained** by developers. Human responsibility remains essential for ethical software development.