

**NATIONAL TEXTILE**

**UNIVERSITY**

DEPARTMENT OF COMPUTER SCIENCE

**SUBMITTED BY:**

Farah Naz 23-NTU-CS-1152

**SECTION SE: 5th(A)**

**LAB MANUAL**

**SUBMITTED TO:**

Sir Nasir Mehmood

**SUBMISSION DATE:**

10-10-2025

**Question 1:**



**Create Thread:**

**Code:**

 #include <stdio.h>

 #include <pthread.h>

 #include <unistd.h>

 // Thread function - this will run in the new thread

 void\* thread\_function(void\* arg) {

 printf("Hello from the new thread!\n");

 printf("Thread ID: %lu\n", pthread\_self());

 return NULL;

 }

 int main() {

 pthread\_t thread\_id;

 printf("Main thread starting...\n");

 printf("Main Thread ID: %lu\n", pthread\_self());

 // Create a new thread

 pthread\_create(&thread\_id, NULL, thread\_function, NULL);

// Wait for the thread to finish

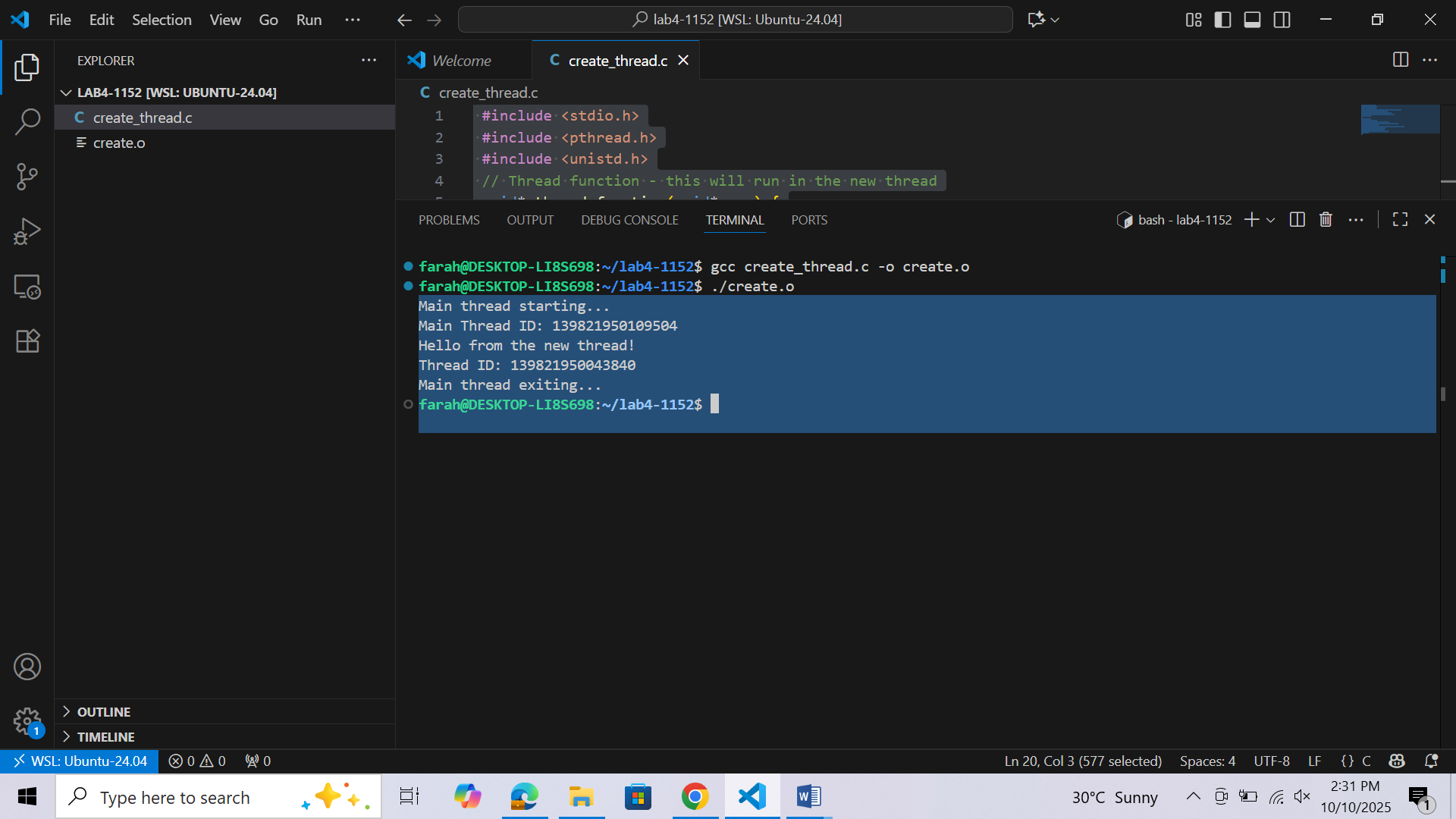
 pthread\_join(thread\_id, NULL);

 printf("Main thread exiting...\n");

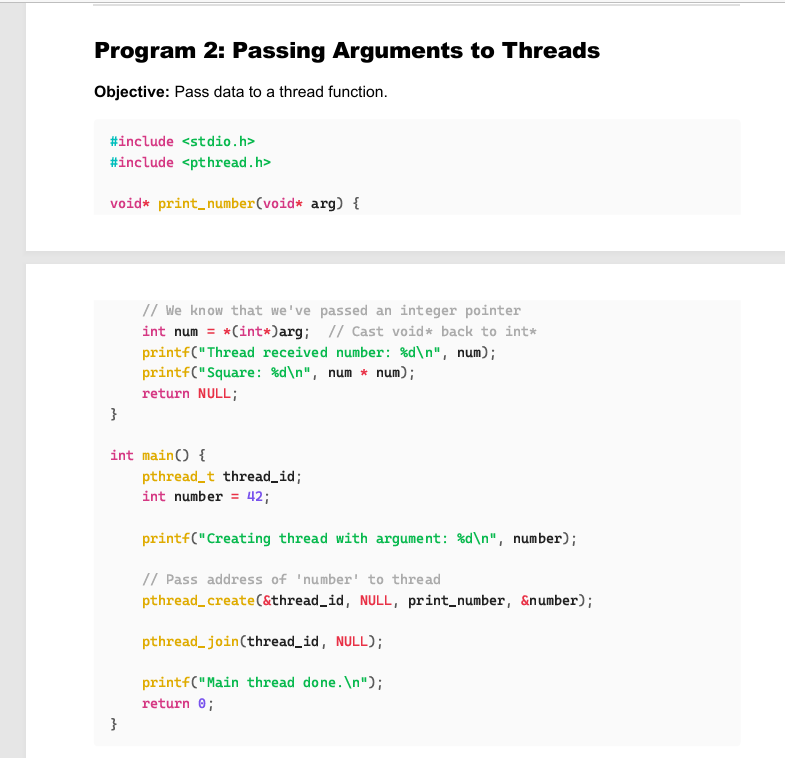
 return 0;

 }

**Output:**



**Question 2:**



**Pass Arguments to the thread:**

**Code:**

#include <stdio.h>

#include <pthread.h>

    void\* print\_number(void\* arg) {

    // We know that we've passed an integer pointer

    int num = \*(int\*)arg; // Cast void\* back to int\*

    printf("Thread received number: %d\n", num);

    printf("Square: %d\n", num \* num);

    return NULL;

 }

 int main() {

    pthread\_t thread\_id;

    int number = 42;

    printf("Creating thread with argument: %d\n", number);

    // Pass address of 'number' to thread

    pthread\_create(&thread\_id, NULL, print\_number, &number);

    ;pthread join will block main process until the created one completes and then resume the main thread

    pthread\_join(thread\_id, NULL);

    printf("Main thread done.\n");

    return 0;

 }

Output:

