BAHRIA UNIVERSITY (KARACHI CAMPUS)

OPEN ENDED LAB II – **Fall22**

(System Programing (LAB) **CSC-454)**

Class: BSE [4]-5 (B) (Morning)

Course Instructor: **Engr Rizwan Fazal / Engr Rehan Baig**

Time Allowed: **1.5 Hour**

Max Marks: **6**

Student's Name: Uswa Asif

Reg. No: 69989

Enrollment : 02-131202-038

Instructions:

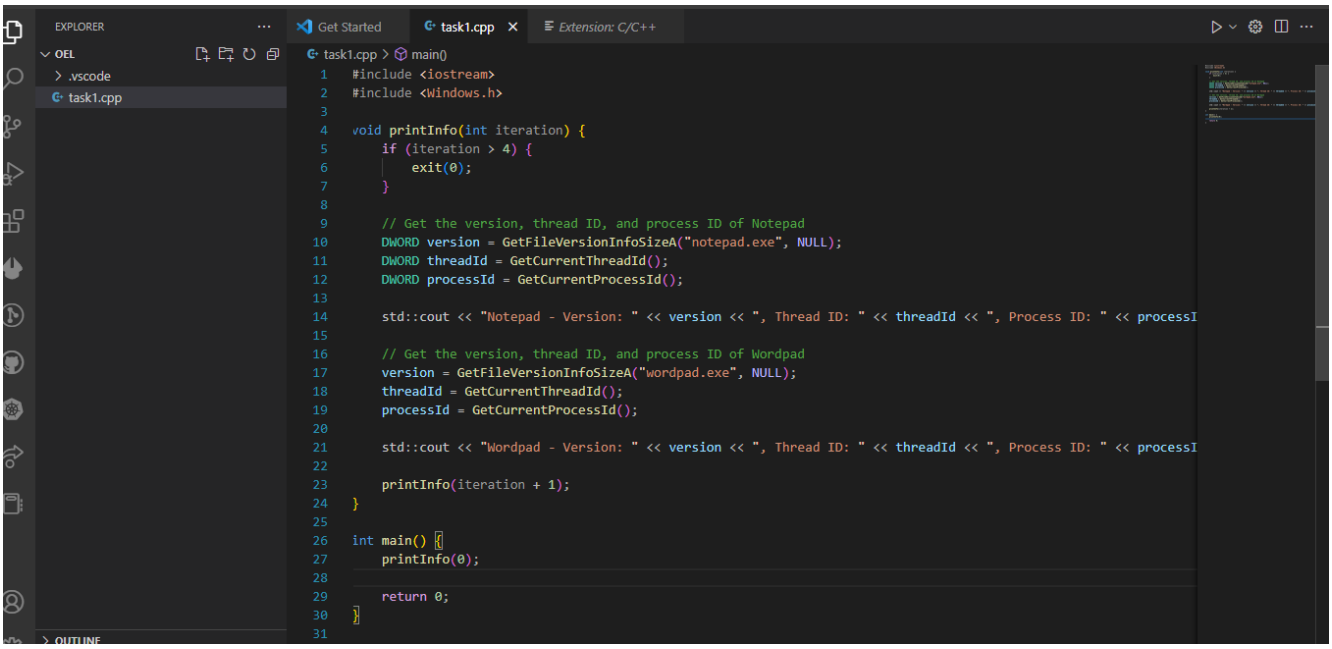
1. Submit your answers within file against each question with screenshot of both code and solution output.
2. File must be submitted in .pdf.

[CLO#05, 6 marks]

SCENARIO:

You are working as a system engineer in a Microsoft vendor company that creates Apps for Microsoft store.
Your Project manager assigned you a task to design an application for code editor for Microsoft store. For that you need to analyze the basics of NotePad/WordPad applications that comes built-in with Microsoft windows. You need to create a process and analyze the following for notepad and WordPad.

Q1: Run a loop or Use Recursion which enable program to print 5 times following for both Notepad and WordPad **(versionId, ThreadId, processId)**, meanwhile use exit thread function that-should be interrupt when counter reaches on 4rth iteration. (**4 Marks**)



```cpp
#include <iostream>
#include <Windows.h>

void printInfo(int iteration) {
    if (iteration > 4) {
        exit(0);
    }

    // Get the version, thread ID, and process ID of Notepad
    DWORD version = GetFileVersionInfoSizeA("notepad.exe", NULL);
    DWORD threadId = GetCurrentThreadId();
    DWORD processId = GetCurrentProcessId();

    std::cout << "Notepad - Version: " << version << ", Thread ID: " << threadId << ", Process ID: " << processI

    // Get the version, thread ID, and process ID of Wordpad
    version = GetFileVersionInfoSizeA("wordpad.exe", NULL);
    threadId = GetCurrentThreadId();
    processId = GetCurrentProcessId();

    std::cout << "Wordpad - Version: " << version << ", Thread ID: " << threadId << ", Process ID: " << processI

    printInfo(iteration + 1);
}

int main() {
    printInfo(0);

    return 0;
}
```

```
1:
versionId = Notepad, ThreadId = 140126619633408, ProcessId = 3753
2:
versionId = WordPad, ThreadId = 140126619633408, ProcessId = 3753
3:
versionId = Notepad, ThreadId = 140126619633408, ProcessId = 3753
4:
versionId = WordPad, ThreadId = 140126619633408, ProcessId = 3753
5:
```

Q2: Write a code for any two synchronization objects from following. (**2 Marks**)

1. Events
2. Semaphores
3. Mutexes

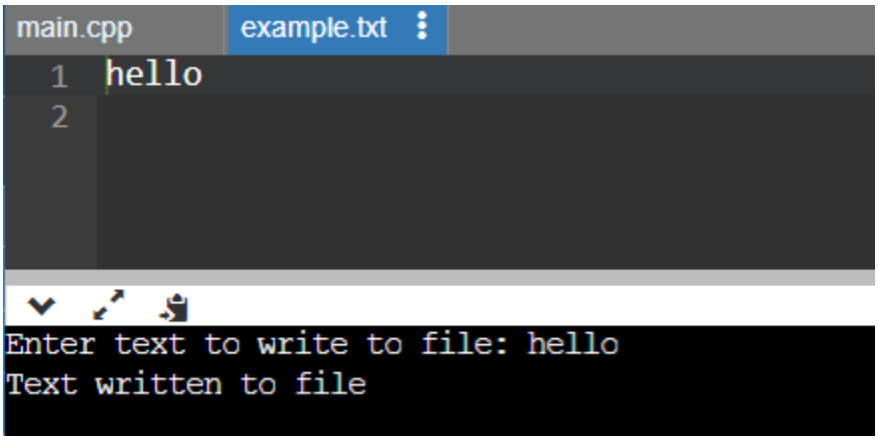
1) Mutex

```cpp
#include <iostream>
#include <fstream>
#include <mutex>
using namespace std;

// Declare a global mutex object
std::mutex file_mutex;

void writeToFile(string text) {
    // Lock the mutex before accessing the file
    std::unique_lock<std::mutex> lock(file_mutex);
    ofstream myfile;
    myfile.open("example.txt", ios::app);
    myfile << text << endl;
    myfile.close();
    // Release the lock
    lock.unlock();
}

int main() {
    cout << "Enter text to write to file: ";
    string text;
    cin >> text;
    writeToFile(text);
    cout << "Text written to file" << endl;
    return 0;
}
```



2) Semaphores

```cpp
#include <iostream>
#include <fstream>
#include <semaphore.h>
using namespace std;

sem_t sem;

void writeToFile(string text) {
    sem_wait(&sem);
    ofstream myfile;
    myfile.open("example.txt", ios::app);
    myfile << text << endl;
    myfile.close();
    sem_post(&sem);
}

int main() {
    sem_init(&sem, 0, 1);
    cout << "Enter text to write to file: ";
    string text;
    cin >> text;
    writeToFile(text);
    sem_destroy(&sem);
    return 0;
}
```

main.cpp example.txt

```
uswa
```

```
Enter text to write to file: uswa
```