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: Jamshed Ali Enrollment : 02-131202-008

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**)

Solution:

```
#include <iostream>
#include <iostream>
#include <thread>
#include <unistd.h>
using namespace std;
void print_info(int iteration) {
 if (iteration == 4) {
 pthread_exit(NULL);
 }
 std::string app_name;
 if(iteration % 2 == 0)
 app_name = "Notepad";
 else
 app_name = "WordPad";
 std::cout << "\nApp Name: " << app_name << ",\n Thread ID: " << std::this_thread::get_id() << ",\n
Process ID: " <<
getpid() << std::endl;
}
int main() {
 std::cout << "Processes Are Given Below:\n" <<std::endl;
 for (int id = 0; id <=4; id++) {
 std::thread put(print_info, id);
 put.join();
 }

 return 0;
}
```

Output:

```
Processes Are Given Below:


App Name: Notepad,
 Thread ID: 22924215375616,
 Process ID: 305

App Name: WordPad,
 Thread ID: 22924215375616,
 Process ID: 305

App Name: Notepad,
 Thread ID: 22924215375616,
 Process ID: 305

App Name: WordPad,
 Thread ID: 22924215375616,
 Process ID: 305
```

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

1. Events
2. Semaphores
3. Mutexes

Solution:

```
#include <iostream>
#include <thread>
#include <mutex>
#include <semaphore.h>
#include<vector>

sem_t semaphore;
std::mutex mutex;
std::vector<int> shared_resource;

void addToResource(int value) {
   sem_wait(&semaphore); // Wait on the semaphore
   {
      std::lock_guard<std::mutex> lock(mutex); // Acquire the mutex lock
      shared_resource.push_back(value); // Access the shared resource
   }
   sem_post(&semaphore); // Release the semaphore
```

```
}

int main() {
   sem_init(&semaphore, 0, 1); // Initialize the semaphore with a value of 1
   std::thread t1(addToResource, 1);
   std::thread t2(addToResource, 2);
   t1.join();
   t2.join();
   std::cout << "Final value of shared resource: [ ";
   for(auto i: shared_resource)
      std::cout << i << " ";
   std::cout << "]" << std::endl;
   sem_destroy(&semaphore);
   return 0;
}
```

Output: