4B OS LAB 8 TASKS

Muhammad Danish | cs182019 |4B

Task-1:

task1.cpp | task2.cpp | task3.cpp | task4.cpp | task5.cpp | task6.cpp | task7.cpp

C:\Users\user\Desktop\Semester 4\OS Lab\Lab 08\Tasks\task1.cpp

```
using namespace std;


void threadFunction()
{
    for(int i=0;i<5;i++)
    {
        cout<< "Thread using function pointer." <<endl;
    }
}

int main()
{
    thread threadObj(threadFunction);
    cout<<"Display from main thead."<<endl;
    threadObj.join();

    cout<<"\nExit of main function."<<endl;
    return 0;
}
```

```
Display from main thead.Thread using function pointer.
Thread using function pointer.
Thread using function pointer.
Thread using function pointer.
Thread using function pointer.


Exit of main function.

--------------------------------
Process exited after 0.06582 seconds with return value 0
Press any key to continue . . .
```

(Continued…)

Task-2:

task1.cpp | task2.cpp | task3.cpp | task4.cpp | task5.cpp | task6.cpp | task7.cpp | q1.cpp

```cpp
#include <iostream>
#include <thread>
using namespace std;
class DisplayThread
{
    public:
        void operator()()
        {
            for(int i=0;i<5;i++)
            {
                cout<<"Thread using fucntion object."<<endl;
            }
        }
};

int main()
{
    thread threadObj( ( DisplayThread() ) );
    for(int i=0;i<5;i++)
    {
        cout<<"Display from main thread."<<endl;
    }

    cout<<"Waiting for thread to complete."<<endl;
    threadObj.join();

    cout<<"Exiting from main thread"<<endl;
    return 0;
}
```

```
Display from main thread.Thread using fucntion object.
Thread using fucntion object.
Thread using fucntion object.
Thread using fucntion object.
Thread using fucntion object.

Display from main thread.
Display from main thread.
Display from main thread.
Display from main thread.
Waiting for thread to complete.
Exiting from main thread

--------------------------------
Process exited after 0.07343 seconds with return value 0
Press any key to continue . . .
```

(Continued…)

Task-3:

task1.cpp | task2.cpp | task3.cpp | task4.cpp | task5.cpp | task6.cpp | task7.cpp | q1.

```cpp
#include <iostream>
#include <thread>

using namespace std;
int main()
{
    int x =9;
    thread threadObj([]{
        for(int i=0;i<5;i++)
        {
            cout <<"thread using lambda fucntion."<<endl;
        }
    });

    for(int i=0;i<5;i++)
    {
        cout<<"Display from main thread."<<endl;
    }
    threadObj.join();
    cout<<"Exiting from main thread."<<endl;
    return 0;
}
```

```
Display from main thread.thread using lambda fucntion.
thread using lambda fucntion.
thread using lambda fucntion.
thread using lambda fucntion.
thread using lambda fucntion.

Display from main thread.
Display from main thread.
Display from main thread.
Display from main thread.
Exiting from main thread.

--------------------------------
Process exited after 0.1314 seconds with return value 0
Press any key to continue . . .
```

(Continued…)

Task-4:

task1.cpp | task2.cpp | task3.cpp | task4.cpp | task5.cpp | task6.cpp | task7.c

```cpp
#include <iostream>
#include <string>
#include <thread>
using namespace std;
class DisplayThread
{
    public:
        void operator()(int x,string str){
            cout<<"Passed Number = "<<x<<endl;
            cout<<"Passed String = "<<str<<endl;
        }
};

void threadFunction(int x,string str){
            cout<<"Passed Number = "<<x<<endl;
            cout<<"Passed String = "<<str<<endl;
}

int main(){
    int x=10;
    string str ="SAMPLE STRING.";
    thread threadObj1(threadFunction,x,str);
    thread threadObj2((DisplayThread()),x,str);
    thread threadObj3([](int x,string str){
        cout<<"Passed Number = "<<x<<endl;
        cout<<"Passed String = "<<str<<endl;
    },x,str);

    threadObj1.join();
    threadObj2.join();
    threadObj3.join();
}
```

```
Passed Number = Passed Number = 10
Passed String = SAMPLE STRING.
10
Passed String = SAMPLE STRING.
Passed Number = 10
Passed String = SAMPLE STRING.

--------------------------------
Process exited after 0.2331 seconds with return value 0
Press any key to continue . . .
```

(Continued…)

Task-5:

```cpp
#include <iostream>
#include <thread>
using namespace std;

void thread_function()
{
    cout<<endl<<"Inside Thread :: ID = "<<this_thread::get_id()<<endl;
}

int main()
{
    thread threadObj1(thread_function);
    thread threadObj2(thread_function);
    if(threadObj1.get_id()!= threadObj2.get_id())
    {
        cout<<endl<<"Both threads have different IDs"<<endl;
    }
    cout<<"From main thread :: ID of thread 1 = "<<threadObj1.get_id()<<endl;
    cout<<"From main thread :: ID of thread 2 = "<<threadObj2.get_id()<<endl;
    threadObj1.join();
    threadObj2.join();
    return 0;

}
```

```
Both threads have different IDs
From main thread :: ID of thread 1 = Inside Thread :: ID = 3
Inside Thread :: ID = 2
2
From main thread :: ID of thread 2 = 3

--------------------------------
Process exited after 0.08208 seconds with return value 0
Press any key to continue . . .
```

(Continued…)

Task-6:

ask1.cpp | task2.cpp | task3.cpp | task4.cpp | task5.cpp | task6.cpp | tas

```cpp
#include <thread>
#include <iostream>
using namespace std;
class Summation{
    private:
        int sum;
        int limit;
    public:
        Summation(int lim){
            limit=lim;
            sum=0;
        }

        int getSum() const{
            return sum;
        }

        void operator()(){
            for (int i=1;i<=limit;i++)
            {
                sum =sum+i;
            }
        }
};
int main(){
    Summation sumHelper(10);
    thread thread(ref(sumHelper));
    thread.join();
    cout<<"Sum : "<<sumHelper.getSum()<<endl;
    return 0;
}
```

```
Sum : 55

--------------------------------
Process exited after 0.06646 seconds with return value 0
Press any key to continue . . .
```

(Continued…)

Task-7:

task1.cpp | task2.cpp | task3.cpp | task4.cpp | task5.cpp | task6.cpp | tasl

```cpp
#include<iostream>
#include<thread>
using namespace std;
class Summation{
    private:
        int sum;
        int limit;
    public:
        Summation(int lim){
            limit=lim;
            sum=0;
        }
        int getSum() const{
            return sum;
        }

        void operator ()(){
            for(int i=1;i<=limit;i++){
                sum =sum+i;
            }
        }
};
int main()
{
    Summation sumHelper(10);
    thread thread(ref(sumHelper));
    thread.detach();
    cout<<"Sum : "<<sumHelper.getSum()<<endl;
    return 0;
}
```

```
Sum : 0

--------------------------------
Process exited after 0.1276 seconds with return value 0
Press any key to continue . . .
```