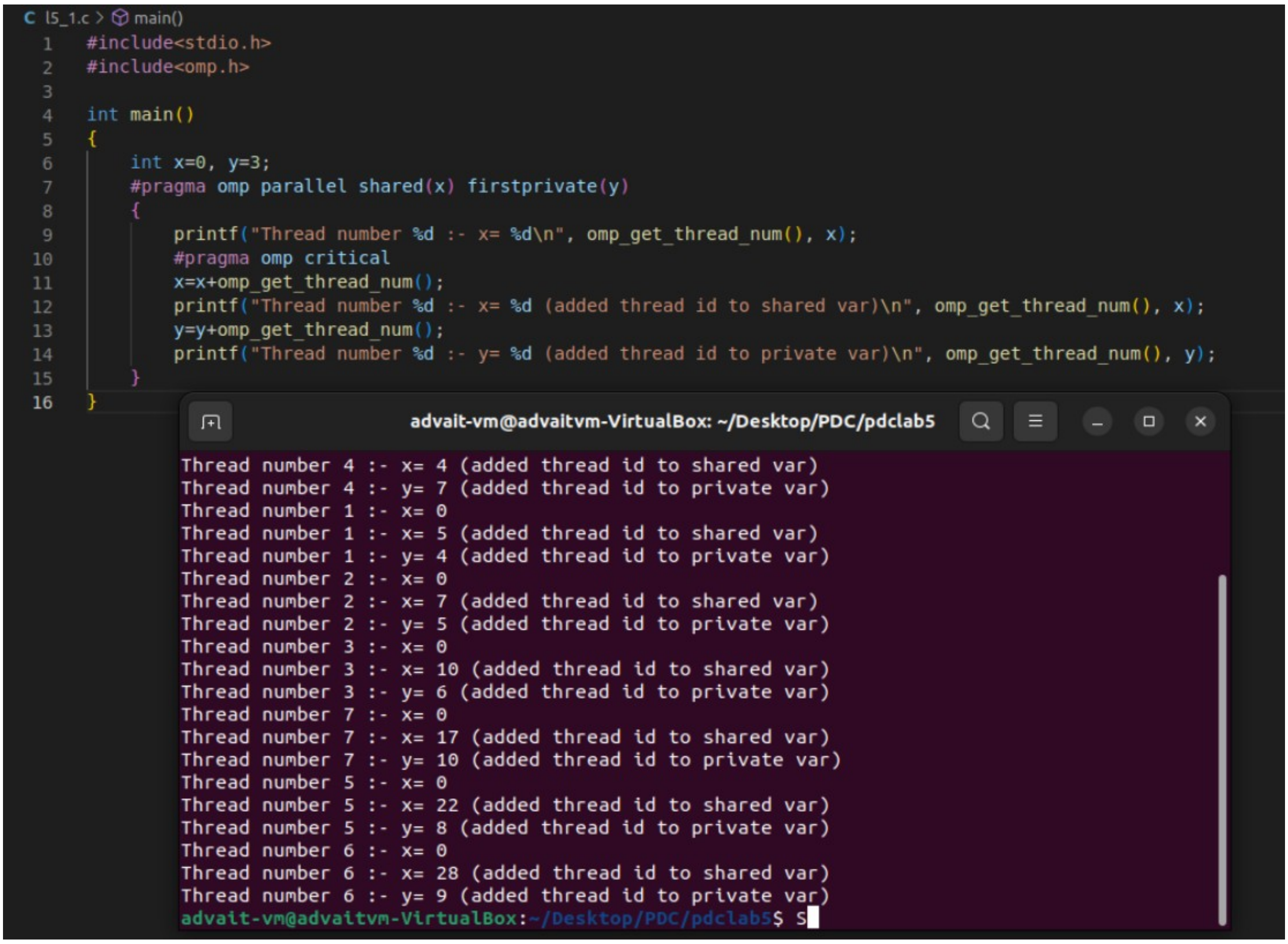
PDC Lab 5 Report

Advait Deochakke
20BCE1143

Example program with x=x+thread_id for critical, single and master. To prove concept, use one shared variable for each synchronization construct.

1) Example Program - Critical

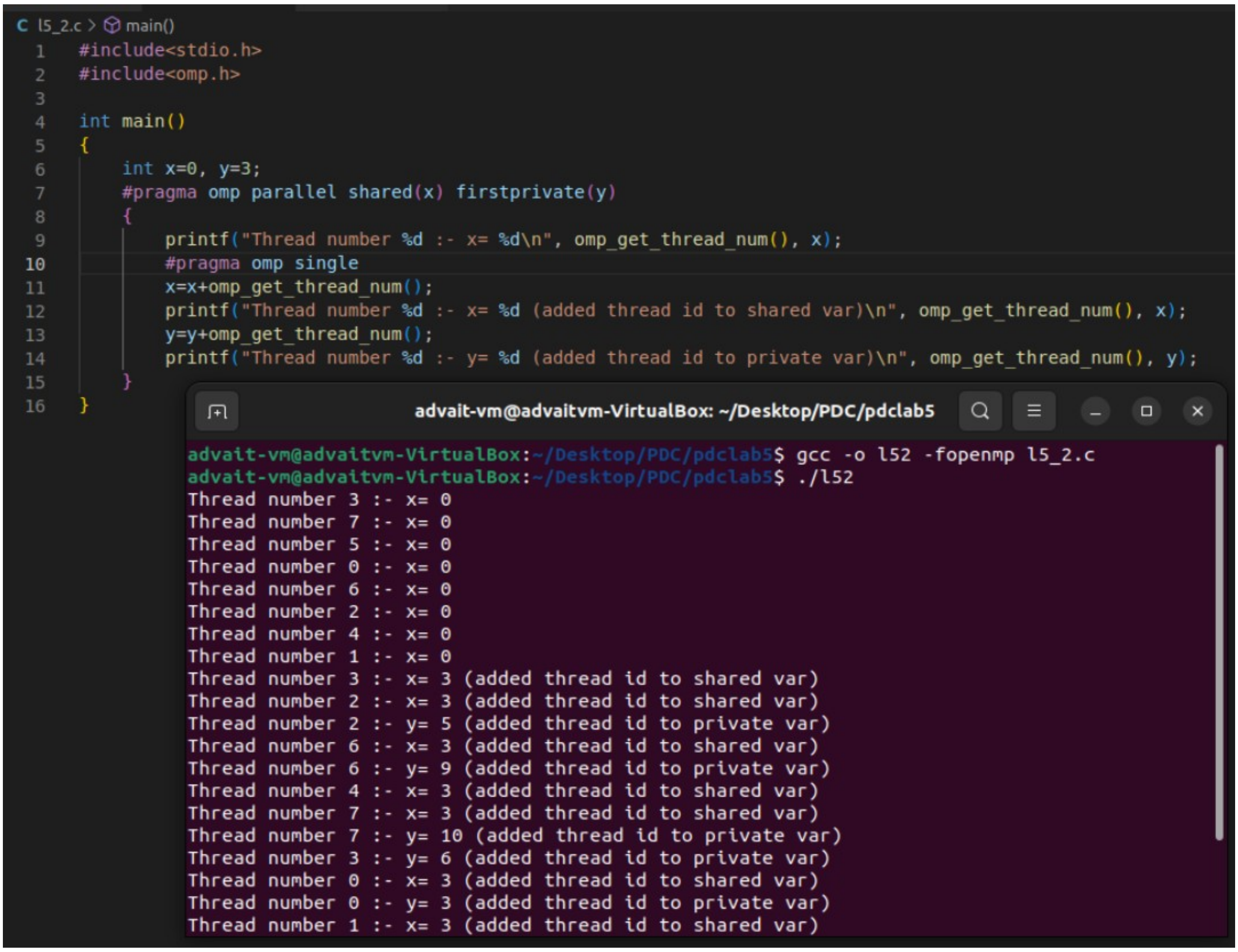


```c
#include<stdio.h>
#include<omp.h>

int main()
{
    int x=0, y=3;
    #pragma omp parallel shared(x) firstprivate(y)
    {
        printf("Thread number %d :- x= %d\n", omp_get_thread_num(), x);
        #pragma omp critical
        x=x+omp_get_thread_num();
        printf("Thread number %d :- x= %d (added thread id to shared var)\n", omp_get_thread_num(), x);
        y=y+omp_get_thread_num();
        printf("Thread number %d :- y= %d (added thread id to private var)\n", omp_get_thread_num(), y);
    }
}
```

```
Thread number 4 :- x= 4 (added thread id to shared var)
Thread number 4 :- y= 7 (added thread id to private var)
Thread number 1 :- x= 0
Thread number 1 :- x= 5 (added thread id to shared var)
Thread number 1 :- y= 4 (added thread id to private var)
Thread number 2 :- x= 0
Thread number 2 :- x= 7 (added thread id to shared var)
Thread number 2 :- y= 5 (added thread id to private var)
Thread number 3 :- x= 0
Thread number 3 :- x= 10 (added thread id to shared var)
Thread number 3 :- y= 6 (added thread id to private var)
Thread number 7 :- x= 0
Thread number 7 :- x= 17 (added thread id to shared var)
Thread number 7 :- y= 10 (added thread id to private var)
Thread number 5 :- x= 0
Thread number 5 :- x= 22 (added thread id to shared var)
Thread number 5 :- y= 8 (added thread id to private var)
Thread number 6 :- x= 0
Thread number 6 :- x= 28 (added thread id to shared var)
Thread number 6 :- y= 9 (added thread id to private var)
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab5$ 
```

2) Example Program - Single

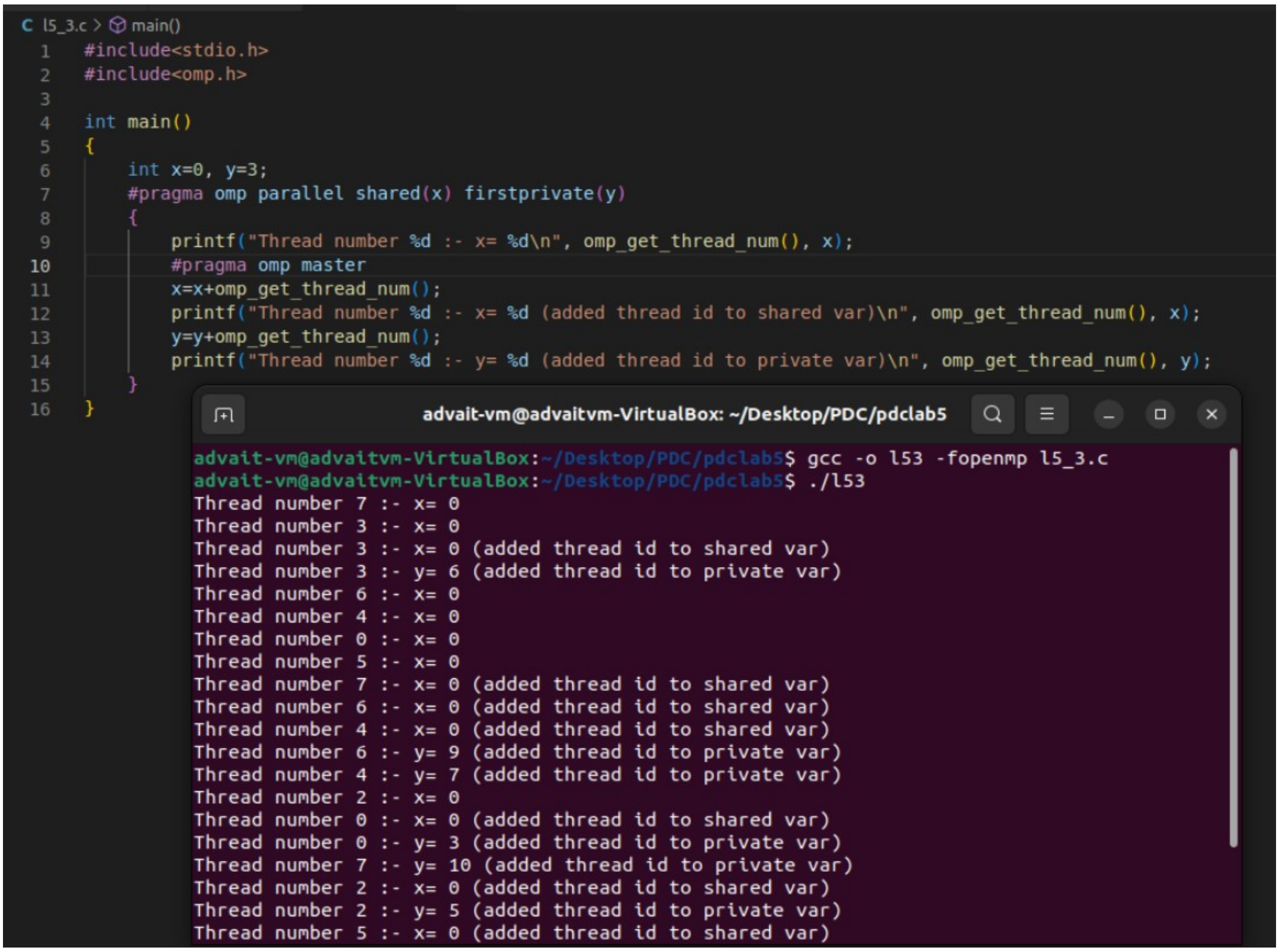


```c
#include<stdio.h>
#include<omp.h>

int main()
{
    int x=0, y=3;
    #pragma omp parallel shared(x) firstprivate(y)
    {
        printf("Thread number %d :- x= %d\n", omp_get_thread_num(), x);
        #pragma omp single
        x=x+omp_get_thread_num();
        printf("Thread number %d :- x= %d (added thread id to shared var)\n", omp_get_thread_num(), x);
        y=y+omp_get_thread_num();
        printf("Thread number %d :- y= %d (added thread id to private var)\n", omp_get_thread_num(), y);
    }
}
```

```
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab5$ gcc -o l52 -fopenmp l5_2.c
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab5$ ./l52
Thread number 3 :- x= 0
Thread number 7 :- x= 0
Thread number 5 :- x= 0
Thread number 0 :- x= 0
Thread number 6 :- x= 0
Thread number 2 :- x= 0
Thread number 4 :- x= 0
Thread number 1 :- x= 0
Thread number 3 :- x= 3 (added thread id to shared var)
Thread number 2 :- x= 3 (added thread id to shared var)
Thread number 2 :- y= 5 (added thread id to private var)
Thread number 6 :- x= 3 (added thread id to shared var)
Thread number 6 :- y= 9 (added thread id to private var)
Thread number 4 :- x= 3 (added thread id to shared var)
Thread number 7 :- x= 3 (added thread id to shared var)
Thread number 7 :- y= 10 (added thread id to private var)
Thread number 3 :- y= 6 (added thread id to private var)
Thread number 0 :- x= 3 (added thread id to shared var)
Thread number 0 :- y= 3 (added thread id to private var)
Thread number 1 :- x= 3 (added thread id to shared var)
```

3) Example Program - Master

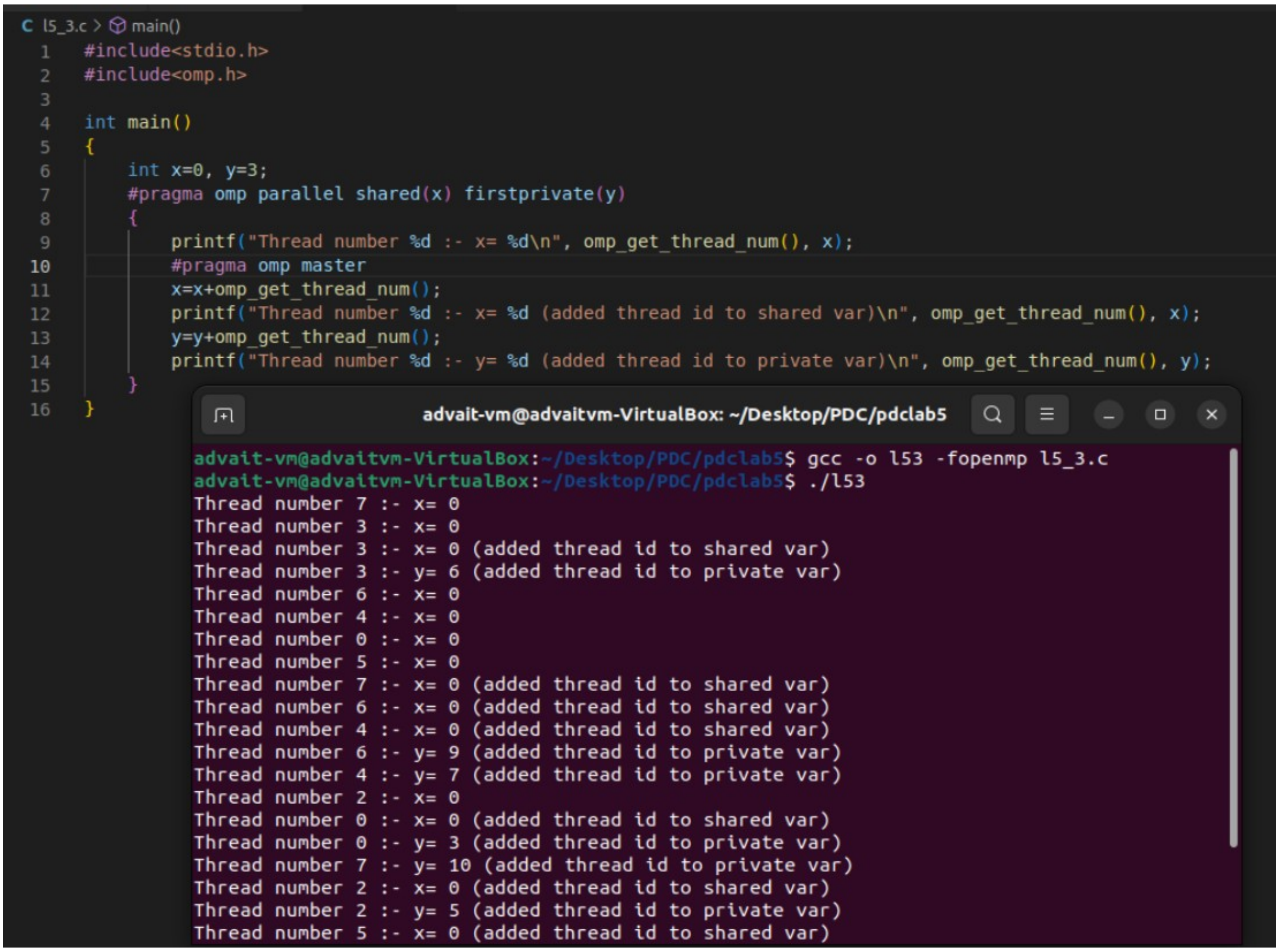


```c
#include<stdio.h>
#include<omp.h>

int main()
{
    int x=0, y=3;
    #pragma omp parallel shared(x) firstprivate(y)
    {
        printf("Thread number %d :- x= %d\n", omp_get_thread_num(), x);
        #pragma omp master
        x=x+omp_get_thread_num();
        printf("Thread number %d :- x= %d (added thread id to shared var)\n", omp_get_thread_num(), x);
        y=y+omp_get_thread_num();
        printf("Thread number %d :- y= %d (added thread id to private var)\n", omp_get_thread_num(), y);
    }
}
```

```
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab5$ gcc -o l53 -fopenmp l5_3.c
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab5$ ./l53
Thread number 7 :- x= 0
Thread number 3 :- x= 0
Thread number 3 :- x= 0 (added thread id to shared var)
Thread number 3 :- y= 6 (added thread id to private var)
Thread number 6 :- x= 0
Thread number 4 :- x= 0
Thread number 0 :- x= 0
Thread number 5 :- x= 0
Thread number 7 :- x= 0 (added thread id to shared var)
Thread number 6 :- x= 0 (added thread id to shared var)
Thread number 4 :- x= 0 (added thread id to shared var)
Thread number 6 :- y= 9 (added thread id to private var)
Thread number 4 :- y= 7 (added thread id to private var)
Thread number 2 :- x= 0
Thread number 0 :- x= 0 (added thread id to shared var)
Thread number 0 :- y= 3 (added thread id to private var)
Thread number 7 :- y= 10 (added thread id to private var)
Thread number 2 :- x= 0 (added thread id to shared var)
Thread number 2 :- y= 5 (added thread id to private var)
Thread number 5 :- x= 0 (added thread id to shared var)
```

VIT placement cell where 100 students are placed in 4 companies. Get as input the name, register number, the pay package of students selected for jobs in the particular organization
Display the total number of students selected in each company.
Display the average pay package of the 100 students.



```
#include<stdio.h>
#include<omp.h>

int main()
{
    int x=0, y=3;
    #pragma omp parallel shared(x) firstprivate(y)
    {
        printf("Thread number %d :- x= %d\n", omp_get_thread_num(), x);
        #pragma omp master
        x=x+omp_get_thread_num();
        printf("Thread number %d :- x= %d (added thread id to shared var)\n", omp_get_thread_num(), x);
        y=y+omp_get_thread_num();
        printf("Thread number %d :- y= %d (added thread id to private var)\n", omp_get_thread_num(), y);
    }
}
```

```
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab5$ gcc -o l53 -fopenmp l5_3.c
advait-vm@advaitvm-VirtualBox:~/Desktop/PDC/pdclab5$ ./l53
Thread number 7 :- x= 0
Thread number 3 :- x= 0
Thread number 3 :- x= 0 (added thread id to shared var)
Thread number 3 :- y= 6 (added thread id to private var)
Thread number 6 :- x= 0
Thread number 4 :- x= 0
Thread number 0 :- x= 0
Thread number 5 :- x= 0
Thread number 7 :- x= 0 (added thread id to shared var)
Thread number 6 :- x= 0 (added thread id to shared var)
Thread number 4 :- x= 0 (added thread id to shared var)
Thread number 6 :- y= 9 (added thread id to private var)
Thread number 4 :- y= 7 (added thread id to private var)
Thread number 2 :- x= 0
Thread number 0 :- x= 0 (added thread id to shared var)
Thread number 0 :- y= 3 (added thread id to private var)
Thread number 7 :- y= 10 (added thread id to private var)
Thread number 2 :- x= 0 (added thread id to shared var)
Thread number 2 :- y= 5 (added thread id to private var)
Thread number 5 :- x= 0 (added thread id to shared var)
```