Practical No.16

Aim: Write a java program for accepting user defined package and accessed the class package in other program.

```
package calculate;

public class calculator

{

public int add(int a, int b)

{

int result=a+b;

System.out.print("additon is=  ");

return a+b;

}

public int mult(int a, int b)

{

System.out.print("multiplication is=  ");

return a*b;

}

public int sub(int a, int b)

{

System.out.print("subtraction is=  ");

return a-b;

}

public int div(int a, int b)

{

System.out.print("division is=  ");

return a/b;
```

```
}
public static void main(String args[])
{
calculator obj =new calculator();
 System.out.println(obj.add(20,40));
 System.out.println(obj.mult(20,40));
 System.out.println(obj.sub(20,40));
 System.out.println(obj.div(60,30));
}
}
```

Output:

```
C:\Users\mahaj\OneDrive\Desktop\java alpha\calculate>javac -d . calculator.java

C:\Users\mahaj\OneDrive\Desktop\java alpha\calculate>java calculate.calculator
additon is=  60
multiplication is=  800
subtraction is=  -20
division is=  2
```

Accesing package in other program:

```
import calculate.calculator;
class accesspkg
{
public static void main(String args[])
{
 calculator obj =new calculator();

 System.out.println(obj.add(50,20));
 System.out.println(obj.mult(5,20));
 System.out.println(obj.sub(70,20));
 System.out.println(obj.div(60,20));
 }
 }
```

Output:

```
C:\Users\mahaj\OneDrive\Desktop\java alpha>jav
additon is=  70
multiplication is=  100
subtraction is=  50
division is=  3
```

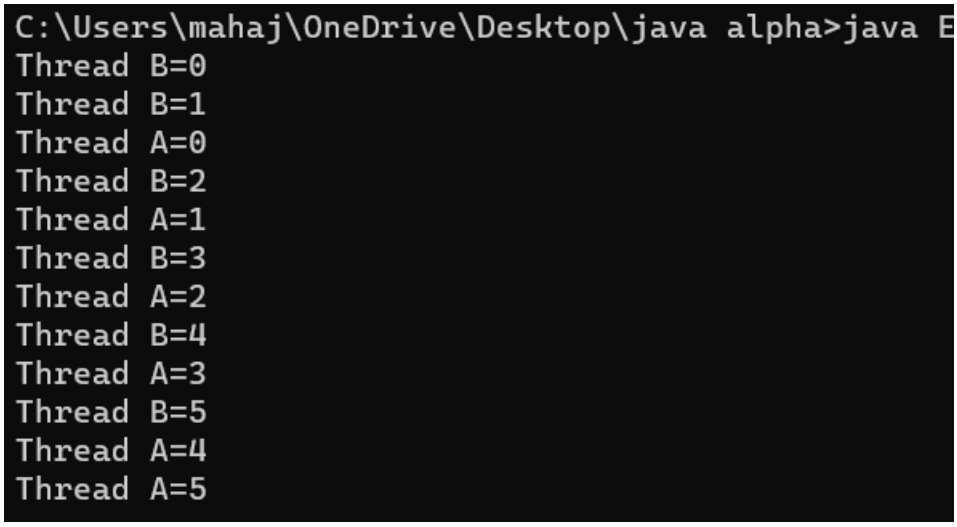
Practical No.17

Aim: write a java program to demonstrate use of threads by:

A) implementing runnable interface .

```
class A implements Runnable
{
public void run()
{
int i;
for(i=0;i<=5;i++)
System.out.println("Thread A" + "=" + i);
}
}
class B implements Runnable
{
public void run()
{
int i;
for(i=0;i<=5;i++)
System.out.println("Thread B" + "=" + i);
}
}
public class ExampleT
{
public static void main(String [] args)
{
Thread t1 = new Thread(new A());
Thread t2 = new Thread(new B()); t1.start();
t2.start();
}
}
```

Output:

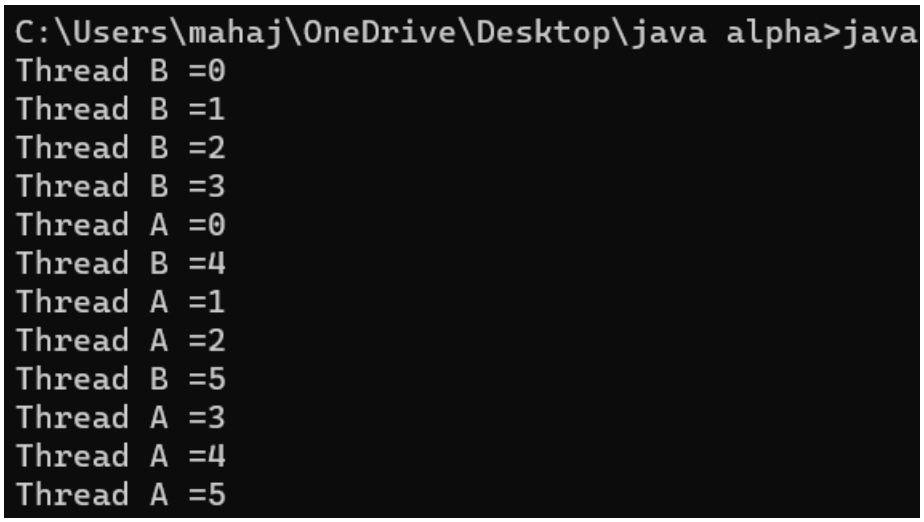


B). by extending thread class

```
class A extends Thread
{
public void run()
{
int i;
for(i=0;i<=5;i++)
System.out.println("Thread A" +" " + "=" + i);
}
 }
class B extends Thread
{
public void run()
{
int i;
for(i=0;i<=5;i++)
System.out.println("Thread B" + " "+ "=" + i);
}
}
```

```
public class Example
{
public static void main(String [] args)
{
A o1 = new A(); B o2 = new B();
o1.start();
o2.start();
}
}
```

OUTPUT:



Practical No.18

Aim: Write a program to demonstrate, suspend() , resume() ,stop() , method of a thread.

```
class threadmethods implements Runnable{
Thread th;
boolean suspend_flag,stop_flag;
 threadmethods(String tN){
th=new Thread(this ,tN);
suspend_flag=false;
 stop_flag=false;
 th.start();
}
public void run(){
try
{
int j=1;
 while(++j<20){
synchronized(this){
while(suspend_flag){
wait();
}
if(stop_flag);
{break;
}}}}
catch(InterruptedException IE){
System.out.println("Thread interrrupted");
}
}
synchronized void my_suspend()
{suspend_flag=true;}
 synchronized void my_resume()
{suspend_flag=false;notify();}
synchronized void my_stop()
{suspend_flag=false;stop_flag=true;notify();}}
```

```
class tmethod{
public static void main(String args[]){
try
{
threadmethods t=new threadmethods("SRS");
System.out.println("Thread t is created & started");
Thread.sleep(2000);
t.my_suspend();
System.out.println("Thread t suspended");
Thread.sleep(2000);
t.my_resume();
System.out.println("Thread t is resumed");
Thread.sleep(2000);
t.my_suspend();
System.out.println("Thread t suspended");
Thread.sleep(2000);
t.my_resume();
System.out.println("Thread t is resumed");
 Thread.sleep(2000);
System.out.println("Thread t is stopped");
}
catch(InterruptedException IE){
System.out.println("Genrated interrupted exception");
}
}
}
```

Output:

```
C:\Users\mahaj\OneDrive\Desktop\java alpha>javac tmet

C:\Users\mahaj\OneDrive\Desktop\java alpha>java tmeth
Thread t is created & started
Thread t suspended
Thread t is resumed
Thread t suspended
Thread t is resumed
Thread t is stopped
```

Practicle No.19

Aim: Write a java program to demonstrate Yield() , sleep() , stop() methods.

```
class A extends Thread{
public void run()
{
for(int i=1;i<=5;i++)
{
if(i==1)
Thread.yield();
System.out.println("\t From Thread A: i= " +i);
}
System.out.println("Exit from A");
}
}
class B extends Thread{
public void run(){
for(int j=1;j<=5;j++)
{
System.out.println("\t From Thread B: j= " +j);
}
System.out.println("Exit from B");
}
}
class C extends Thread{
public void run(){
for(int k=1;k<=5;k++)
{
System.out.println("\t From Thread C: k= " +k);
if(k==1)
try
{
sleep(1000);
}
catch(Exception e)
```

```
{ }
}
System.out.println("Exit from C");
}
}
class ThreadMethods1{
public static void main(String args[]){
A threadA=new A();
B threadB=new B();
C threadC=new C();
System.out.println("Start thread A");
threadA.start();
 System.out.println("Start thread B");
 threadB.start();
 System.out.println("Start thread C");
 threadC.start();
System.out.println("End of main thread");
}
}
```

OUTPUT:

```
PS C:\Users\mahaj\OneDrive\Desktop\java alpha> cd "c:\U
readMethods1.java } ; if ($?) { java ThreadMethods1 }
Start thread A
Start thread B
Start thread C
End of main thread
          From Thread A: i= 1
          From Thread A: i= 2
          From Thread C: k= 1
          From Thread B: j= 1
          From Thread B: j= 2
          From Thread B: j= 3
          From Thread B: j= 4
          From Thread B: j= 5
Exit from B
          From Thread A: i= 3
          From Thread A: i= 4
          From Thread A: i= 5
Exit from A
          From Thread C: k= 2
          From Thread C: k= 3
          From Thread C: k= 4
          From Thread C: k= 5
Exit from C
```

Practical No. 20
Aim:Write a java program to demonstrate thread priorities.

```
import java.lang.*;
public class priorityeg extends Thread{
public void run(){
System.out.println("Now, inside the run method");
}
public static void main(String[]args){
priorityeg th1 = new priorityeg();
 priorityeg th2 = new priorityeg();
System.out.println("The thread priority of first thread is : " + th1.getPriority());
System.out.println("The thread priority of first thread is : " +th2.getPriority());
th1.setPriority(5);
 th2.setPriority(3);
System.out.println("The thread priority of first thread is : " +th1.getPriority());
System.out.println("The thread priority of first thread is : " + th2.getPriority());
System.out.println("The thread priority of main thread is
: " +
Thread.currentThread().getPriority());
Thread.currentThread().setPriority(10);
 System.out.println("The thread priority of main thread is
: " +Thread.currentThread().getPriority());
}
}
```

Output: