V.T Patel Department of Electronics & CommunicationEngineering

Part-IX
MultiThreading

Practical No: 38

AIM: Write a program to create thread which display "Hello World" message.

    A. by extending Thread class

    B. by using Runnable interface

Solution

Practical38.java

```java
package com.jayshil.javaapp;

public class Practical38 extends Thread{
  public void run(){
     System.out.println("Hello World");
  }


  public static void main(String[] args) {
     Practical38 obj1 = new Practical38();
     obj1.run();

  }
}

class Practical38_part2 implements Runnable{
  public void run(){
     System.out.println("Hello World from Runnable");
  }

  public static void main(String[] args) {
     Practical38_part2 obj2 = new Practical38_part2();
     obj2.run();
  }
}
```

Output

Practical No: 39

AIM: Write a program to perform addition of 1to 100 numbers using 4 threads.
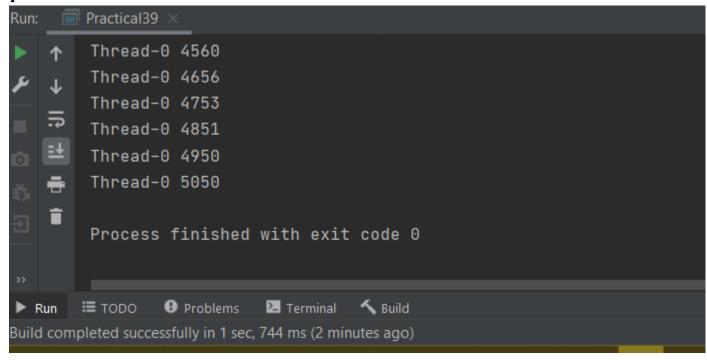

Solution

Practical39.java

```java
package com.jayshil.javaapp;

public class Practical39 extends Thread {
  static int sum = 0,i=0;
  int max = 0;
  public void run(){
     for (; i <= 100; i++) {
         sum =sum + i;
         System.out.println(getName()+" "+sum);

     }

  }

  public static void main(String[] args) {
     Practical39 t1 = new Practical39();
     Practical39 t2 = new Practical39();
     Practical39 t3 = new Practical39();
     Practical39 t4 = new Practical39();
     t1.start();
     try {
        sleep(10);
     } catch (InterruptedException e) {
        e.printStackTrace();
     }
     t2.start();
     t3.start();
     t4.start();

  }
}
```

Output

Run: Practical39 ×

```
Thread-0 4560
Thread-0 4656
Thread-0 4753
Thread-0 4851
Thread-0 4950
Thread-0 5050

Process finished with exit code 0
```

▶ Run   ≔ TODO   ❶ Problems   ⊵ Terminal   ⬩ Build

Build completed successfully in 1 sec, 744 ms (2 minutes ago)

Practical No: 40
AIM: Write a program to increment the value of one variable by one and display it after one second using thread using sleep() method.

Solution

    Practical40.java

```java
package com.jayshil.javaapp;

public class Practical40 extends Thread {
  static int var_value = 5;
  public static void main(String[] args) {


    new Thread(new Runnable() {
       @Override
       public void run() {
          System.out.println("Before increment "+var_value);
          var_value=var_value+1;
       }
    },"Thread 1").start();


    try {
       sleep(1000);
    } catch (InterruptedException e) {
       e.printStackTrace();


    }
    new Thread(new Runnable() {
       @Override
       public void run() {

          System.out.println("Value after increment "+var_value);

       }
    },"Thread 2").start();
  }
}
```

Output:-

```
Run:      Practical40 ×
   D:\softwares\Java\bin\java.exe "-javaagent:D:\softwares\Java intellij\IntelliJ IDEA Community Edition 2020.3.2\lib\idea_rt.jar=
   Before increment 5
   Value after increment 6

   Process finished with exit code 0

▶ Run   ≡ TODO   ❶ Problems   ⊵ Terminal   🔨 Build
Build completed successfully in 1 sec, 542 ms (moments ago)
```

Practical No: 41

AIM: Write a program to create three threads 'FIRST', 'SECOND', 'THIRD'. Set the priority of the 'FIRST' thread to 3, the 'SECOND' thread to 5(default) and the 'THIRD' thread to 7

Solution

Practical41.java

```java
package com.jayshil.javaapp;

public class Practical41 extends Thread{

  public void run(){

     for (int i = 0; i < 10; i++) {
        System.out.println(getName()+" "+i);
     }

  }


  public static void main(String[] args) {
     Practical41 FIRST = new Practical41();
     Practical41 SECOND = new Practical41();
     Practical41 THIRD = new Practical41();

     FIRST.setPriority(3);
     SECOND.setPriority(5);
     THIRD.setPriority(7);

//   FIRST.start();
//   SECOND.start();
//   THIRD.start();

     System.out.println("FIRST thread priority : " + FIRST.getPriority());
     System.out.println("SECOND thread priority : " + SECOND.getPriority());
     System.out.println("THIRD thread priority : " + THIRD.getPriority());
  }
}
```

Output:-

```
Run:    Practical41

    D:\softwares\Java\bin\java.exe "-javaagent:D:\softwares\Java intellij\IntelliJ IDEA Community Edition 2
    FIRST thread priority : 3
    SECOND thread priority : 5
    THIRD thread priority : 7

    Process finished with exit code 0


    Run    ≡ TODO    ❶ Problems    ⊵ Terminal    ↖ Build
Build completed successfully in 1 sec, 542 ms (moments ago)
```

Practical No: 42

AIM: WAP to show the use of Synchronized keyword/ synchronized methods.

Before Synchronization

      Solution:-

            Practical42.java

```java
class BeforeSynchronization {
  void PrintLoop1() {
    for (int i = 0; i < 11; i++) {
      System.out.println(Thread.currentThread().getName() + " : " + i);
      try {
        Thread.sleep(500);
      } catch (InterruptedException e) {
        e.printStackTrace();
      }
    }
  }

  public static void main(String[] args) {
    BeforeSynchronization Object = new BeforeSynchronization();

    new Thread("First Thread"){
      public void run(){
        Object.PrintLoop1();
      }
    }.start();

    new Thread("Second Thread"){
      public void run(){
        Object.PrintLoop1();
      }
    }.start();
  }
}
```

Output:-

```
Run:    BeforeSynchronization ×

   D:\softwares\Java\bin\java.exe "-javaagent:D:\softwares\Java intellij\IntelliJ IDEA Commur
   First Thread : 0
   Second Thread : 0
   First Thread : 1
   Second Thread : 1
   First Thread : 2
   Second Thread : 2
   First Thread : 3
   Second Thread : 3
   First Thread : 4
   Second Thread : 4
   First Thread : 5
   Second Thread : 5
   First Thread : 6
   Second Thread : 6
   First Thread : 7
   Second Thread : 7
   First Thread : 8
   Second Thread : 8
   First Thread : 9
   Second Thread : 9
   First Thread : 10
   Second Thread : 10

   Process finished with exit code 0
```

After Synchronization

Practical42.java

```java
class AfterSynchronization {
  synchronized void PrintLoop2(){
     for (int i = 0; i < 11; i++) {
        System.out.println(Thread.currentThread().getName()+" : "+i);
        try {
          Thread.sleep(500);
        } catch (InterruptedException e) {
          e.printStackTrace();
        }
     }
  }

  public static void main(String[] args) {
     AfterSynchronization Object = new AfterSynchronization();

     new Thread("First Thread"){
        public void run(){
          Object.PrintLoop2();
        }
     }.start();

     new Thread("Second Thread"){
        public void run(){
          Object.PrintLoop2();
        }
     }.start();

  }

}
```

Output:-

```
Run:     AfterSynchronization ×
  ▶   ↑    D:\softwares\Java\bin\java.exe "-javaagent:D:\softwares\Java intellij\IntelliJ IDEA Community
  🔧  ↓    First Thread : 0
  ■   ⇥    First Thread : 1
          First Thread : 2
  📷  ⬇    First Thread : 3
  🖨      First Thread : 4
          First Thread : 5
  🗑      First Thread : 6
          First Thread : 7
  ⊞       First Thread : 8
          First Thread : 9
  📌      First Thread : 10
          Second Thread : 0
          Second Thread : 1
          Second Thread : 2
          Second Thread : 3
          Second Thread : 4
          Second Thread : 5
          Second Thread : 6
          Second Thread : 7
          Second Thread : 8
          Second Thread : 9
          Second Thread : 10

          Process finished with exit code 0


  ▶ Run    ≡ TODO    ❶ Problems    ⏽ Terminal    ↙ Build
```