MODULE 8q - Threaded Code

THE synchronized MODIFIER

In the following program method get() in class SharedObject is qualified by the synchronized modifier:

```
public class ThreadExA
 { public static void main(String[] args)
    { SharedObject pot = new SharedObject();
      ThreadA tA = new ThreadA(pot);
      ThreadB tB = new ThreadB(pot);
      tA.start();
      tB.start();
    }
 }

class SharedObject
 { public synchronized void get(String s, long n)
    { for (int i=0; i<3; i++)
       { System.out.println("Got by " + s);
         try
          { Thread.sleep(n);                 // Pause for a moment
          }
         catch(InterruptedException e)
          {}
       }
    }
 }

class ThreadA extends Thread
 { private SharedObject shob;

   public ThreadA(SharedObject ob)
    { this.shob = ob;
    }

   public void run()
    { for (int i=0; i<2; i++)
         this.shob.get("A", 1000L);
    }
 }

class ThreadB extends Thread
 { private SharedObject shob;

   public ThreadB(SharedObject ob)
    { this.shob = ob;
    }

   public void run()
    { for (int i=0; i<2; i++)
          this.shob.get("B", 100L);
    }
 }

// With the synchronized          With the synchronized
// modifier present the           modifier removed it
// above yields:                  yields:
//
// Got by A                       Got by A
// Got by A                       Got by B
// Got by A                       Got by B
// Got by B                       Got by B
// Got by B                       Got by B
// Got by B                       Got by B
// Got by A                       Got by B
// Got by A                       Got by A
// Got by A                       Got by A
// Got by B                       Got by A
// Got by B                       Got by A
// Got by B                       Got by A
```

TRY IT OUT

Key the above program in, compile it and run it. Then remove the synchronized modifier, recompile, and run the program again. The two sets of results ought to be as shown in the comments at the end of the program.

A MINOR ADDITION TO THE PROGRAM

Next add the try-catch clauses shown in the version below:

```
public class ThreadExB
 { public static void main(String[] args)
    { SharedObject pot = new SharedObject();
      ThreadA tA = new ThreadA(pot);
      ThreadB tB = new ThreadB(pot);
      tA.start();
      tB.start();
    }
 }

class SharedObject
 { public synchronized void get(String s, long n)
    { for (int i=0; i<3; i++)
       { System.out.println("Got by " + s);
         try
          { Thread.sleep(n);                 // Pause for a moment
          }
         catch(InterruptedException e)
          {}
       }
    }
 }

class ThreadA extends Thread
 { private SharedObject shob;

   public ThreadA(SharedObject ob)
    { this.shob = ob;
    }

   public void run()
    { for (int i=0; i<2; i++)
       { this.shob.get("A", 1000L);
```

```
         try                                   // The
          { this.sleep(1000L);                 // only
          }                                    // new
         catch(InterruptedException e)         // code
          {}                                   // added
       }
    }
 }

class ThreadB extends Thread
 { private SharedObject shob;

   public ThreadB(SharedObject ob)
    { this.shob = ob;
    }

   public void run()
    { for (int i=0; i<2; i++)
         this.shob.get("B", 100L);
    }
 }

// This yields:
//
// Got by A
// Got by A
// Got by A
// Got by B
// Got by B
// Got by B
// Got by B
// Got by B
// Got by B
// Got by A
// Got by A
// Got by A
```

TRY IT OUT

This version ought to give the results shown as comments.