

1. Question

10 points

Which amongst the following are methods of the Object class?

- a) notify();
- b) notifyAll();
- c) isInterrupted();
- d) synchronized();
- e) interrupt();
- f) wait(long msecs);
- g) sleep(long msecs);
- h) yield();

- 1. ☐ a, b, d
- 2. ☐ None of these.
- 3. ☐ b, d, e
- 4. ☐ a, b, f
- 5. ☐ b, c, d

Incorrect

2. Question

10 points

Which of the following line of code is suitable to start a thread?

```
class Demo implements Runnable
{
    public void run() {
        System.out.println("Thread is in Running state");
    }

    public static void main(String args[])
    {
        /* Missing code? */
    }
}
```

- 1. ☐

```
Thread t = new Thread(X) ;
t.start() ;
```

- 2. ☐

```
Thread t = new Thread() ;
x.run() ;
```

- 3. ☐

```
X obj = new X() ;
Thread tobj = new Thread(obj) ;
```

```
tobj.start();
```

4. ☐ None of these.
5. ☐ Thread t = new Thread(X);

**Incorrect**

3. Question

**10 points**

**Which method must be defined by a class implementing the java.lang.Runnable interface?**

1. ☐ public void start()
2. ☐ None of these.
3. ☒ public void run()
4. ☐ void run()

**Incorrect**

4. Question

**10 points**

**What will be the output of the below Java program?**

```
package practiceExercise;

class MyThread extends Thread
{
    MyThread() {}
    MyThread(Runnable r) {super(r); }
    public void run()
    {
        System.out.print("Inside Thread ");
    }
}

class RunnableDemo implements Runnable
{
    public void run()
    {
        System.out.print(" Inside Runnable");
    }
}

class ThreadDemo
{
    public static void main(String[] args)
    {
        new MyThread().start();
        new MyThread(new RunnableDemo()).start();
    }
}
```

1. ☐ Throws exception at runtime.

- 2. ☐ Prints "Inside Thread Inside Runnable".
- 3. ☐ Does not compile.
- 4. ☐ Prints "Inside Thread Inside Thread".
- 5. ☐ None of these.

**Incorrect**

5. Question

10 points

**What will be the output of the below Java program?**

```
public class ThreadWaitTest
{
    public static void main(String [] args)
    {
        System.out.print("1 ");
        synchronized(args)
        {
            System.out.print("2 ");
            try
            {
                args.wait(); /* Line 11 */
            }
            catch (InterruptedException e){ }
        }
        System.out.print("3 ");
    }
}
```

- 1. ☐ It fails to compile because the IllegalMonitorStateException of wait() is not dealt with in line 11.
- 2. ☐ 1 2 3
- 3. ☐ 1 2
- 4. ☐ None of these.
- 5. ☐ 1 3

**Incorrect**

6. Question

10 points

**What is the output of the following Java program?**

```
public class ThreadDemo implements Runnable
{
    private int x=0;
    private int y=0;
    public static void main(String [] args)
    {
        ThreadDemo obj = new ThreadDemo();
    }
}
```

```

        (new Thread(obj)).start(); /*Line 8*/
        (new Thread(obj)).start(); /*Line 8*/
    }
    public synchronized void run () /*Line 11*/
    {
        for (int i=0;i<10;i++) /*Line 13*/
        {
            x++;
            y++;
            System.out.println("x = " +x+ " ; y = "+y);
        }
    }
}

```

1. ☐ An error at line 11 causes compilation to fail.
2. ☐ The program prints pairs of values for x and y that are always the same on the same line (e.g. "x=1, y=1". In addition, these values are printed from "x=1; y=1" to "x=20 ; y=20" for both threads.
3. ☐ Errors at lines 8 and 9 cause compilation to fail.
4. ☐ None of these.
5. ☐ The program prints pairs of values for x and y that might not always be the same on the same line (e. g. "x=2, y=1")

**Incorrect**

7. Question

**10 points**

**What will be the output of the following Java program?**

```

package practiceExercise;

class MyThread extends Thread
{
    MyThread()
    {
        System.out.print(" MyThread");
    }
    public void run()
    {
        System.out.print(" bar");
    }
    public void run(String s)
    {
        System.out.println(" baz");
    }
}

public class ThreadDemo
{
    public static void main (String [] args)
    {

```

```

        Thread t = new MyThread()
        {
            public void run()
            {
                System.out.println(" foo");
            }
        };
        t.start();
    }
}

```

1. ☐ MyThread foo
2. ☐ foo
3. ☐ None of these.
4. ☐ foo bar
5. ☐ MyThread bar

**Incorrect**

8. Question

**10 points**

**Which of the following methods make thread leave the running state?**

```

i. yield()
ii. wait()
iii. notify()
iv. notifyAll()
v. sleep(1000)
vi. Thread.join()
vii. Thread.killThread()

```

1. ☐ iv, v and vii
2. ☐ ii, v and vi
3. ☐ i, ii and iv
4. ☐ None of these.
5. ☐ iii, iv and vii

**Incorrect**

9. Question

**10 points**

**Which method registers a thread in a thread scheduler?**

1. ☐ None of these.
2. ☐ register();
3. ☐ start();

4. ☐ construct();

5. ☐ run();

**Incorrect**

10. Question

10 points

**Which of the following methods cannot directly cause a thread to stop executing?**

1. ☐ None of these.

2. ☐ Calling the SetPriority() method on a Thread object.

3. ☐ Calling sleep() method on a Thread object.

4. ☐ Calling the wait() method on an object.

5. ☐ Calling notify() method on an object.

**Incorrect**

11. Question

10 points

**Which of the following are the valid constructors of Thread class?**

i. Thread(Runnable r, String name)

ii. Thread()

iii. Thread(int priority)

iv. Thread(Runnable r, ThreadGroup g)

v. Thread(Runnable r, int priority)

1. ☐ i and iii

2. ☐ ii and iv

3. ☐ None

4. ☐ i and ii

5. ☐ ii and v

**Incorrect**

12. Question

10 points

**What will be the output of the following Java code snippet?**

```
class ThreadDemo extends Thread
{
    final StringBuffer sb1 = new StringBuffer();
    final StringBuffer sb2 = new StringBuffer();

    public static void main(String args[])
    {
        final ThreadDemo h = new ThreadDemo();
    }
}
```

```

new Thread()
{
    public void run()
    {
        synchronized(this)
        {
            h.sb1.append("Java") ;
            h.sb2.append("Thread") ;
            System.out.println(h.sb1) ;
            System.out.println(h.sb2) ;
        }
    }
}.start() ;

new Thread()
{
    public void run()
    {
        synchronized(this)
        {
            h.sb1.append("Multithreading") ;
            h.sb2.append("Example") ;
            System.out.println(h.sb2) ;
            System.out.println(h.sb1) ;
        }
    }
}.start() ;
}
}

```

1. ☐ The order in which text will be printed cannot be determined.
2. ☐ Java Thread ThreadExample JavaMultithreading.
3. ☐ None of these.
4. ☐ Java Thread Multithreading Example.
5. ☐ Multithreading Example java Thread.

**Incorrect**

13. Question

**10 points**

**Assume the following method is properly synchronized and called from a thread A on an object B:**

`wait(2000);`

**After calling this method, when will the thread A become a candidate to get another turn at the CPU?**

1. ☐ After thread A is notified, or after two seconds.
2. ☐ After the lock on B is released, or after two seconds.
3. ☐ Two seconds after thread A is notified.
4. ☐ Two seconds after lock B is released.

5. ☐ None of these.

**Incorrect**

14. Question

10 points

**What is the name of the method that is used to start the execution of a thread?**

1. ☐ start();
2. ☐ resume();
3. ☐ None of these.
4. ☐ init();
5. ☐ run();

**Incorrect**

15. Question

10 points

**What is the output of the following Java code fragment?**

```
class ThreadDemo implements Runnable
{
    int x = 0, y = 0;
    int addX() {x++; return x;}
    int addY() {y++; return y;}

    public void run() {
        for(int i = 0; i < 10; i++)
            System.out.println( Thread.currentThread().getName()
+ ": " +addX() + " " + addY());
    }

    public static void main(String args[])
    {
        ThreadDemo obj1 = new ThreadDemo();
        ThreadDemo obj2 = new ThreadDemo();
        Thread t1 = new Thread(obj1);
        Thread t2 = new Thread(obj2);
        t1.start();
        t2.start();
    }
}
```

1. ☐ None of these.
2. ☐ Will print in this order: 1 1 2 2 3 3 4 4 5 5...
3. ☐ Will print in this order: 1 2 3 4 5 6... 1 2 3 4 5 6...
4. ☐ Compile time Error: There is no start() method.



5. ☐ Will print but may or may not be in an order for Thread t1 and t2 (e.g: 1 1 2 2 1 1 3 3... up to 10 10)

**Incorrect**

16. Question

**10 points**

**What will be the output of the following Java code?**

```
package practiceExercise;

class ThreadDemo extends Thread
{
    public static void main(String [] args)
    {
        ThreadDemo t = new ThreadDemo();
        t.start();
        System.out.print("one. ");
        t.start();
        System.out.print("two. ");
    }
    public void run()
    {
        System.out.print("Thread ");
    }
}
```

- 1. ☐ Compilation fails.
- 2. ☐ An exception occurs at runtime.
- 3. ☐ It prints "Thread one. Thread two.".
- 4. ☐ The output cannot be determined.
- 5. ☐ None of these.

**Incorrect**

17. Question

**10 points**

**What is the output of the following program?**

```
class ThreadTest extends Thread
{
    public void run()
    {
        for(int i = 0; i < 3; i++)
        {
            System.out.println("A");
            System.out.println("B");
        }
    }
}

class ThreadDemo extends Thread
```

```

{
    public void run()
    {
        for(int i = 0; i < 3; i++)
        {
            System.out.println("C");
            System.out.println("D");
        }
    }
    public static void main(String args[])
    {
        ThreadTest t1 = new ThreadTest();
        ThreadDemo t2 = new ThreadDemo();
        t1.start();
        t2.start();
    }
}

```

1. ☐ Compile time Error There is no start() method.
2. ☐ Will print A B C D but we cannot predict the Order in which this will be printed.
3. ☐ Will print in this order ABCD...ABCD...
4. ☐ None of these.
5. ☐ Will print in this order AB CD AB...

**Incorrect**

18. Question

**10 points**

**What will be the output of the following Java code snippet?**

```

class MyThread extends Thread
{
    public static void main(String [] args)
    {
        MyThread t = new MyThread(); /* Line 5 */
        t.run(); /* Line 6 */
    }

    public void run()
    {
        for(int i=1; i < 3; ++i)
        {
            System.out.print(i + "..");
        }
    }
}

```

1. ☐ None of these.

2. ☐ 1..2..3..

3. ☐ 1..2..

4. ☐ This code will not compile due to line 6.

5. ☐ This code will not compile due to line 5.

**Incorrect**

19. Question

10 points

Which of the following piece of code will create and start this thread?

```
public class MyRunnable implements Runnable
{
    public void run()
    {
        // some code here
    }
}
```

1. ☐ new MyRunnable().start();

2. ☐ None of these.

3. ☐ new Thread(new MyRunnable()).start();

4. ☐ new Thread(MyRunnable).run();

5. ☐ new Runnable(MyRunnable).start();

**Incorrect**

20. Question

10 points

What is the output of the below Java code?

```
class ThreadDemo implements Runnable
{
    String x, y;
    public void run()
    {
        for(int i = 0; i < 10; i++)
            synchronized(this)
            {
                x = "Hello";
                y = "Java";
                System.out.print(x + " " + y + " ");
            }
    }
    public static void main(String args[])
    {
        ThreadDemo run = new ThreadDemo ();
        Thread obj1 = new Thread(run);
    }
}
```

```
Thread obj2 = new Thread(run);  
obj1.start();  
obj2.start();  
}  
}
```

1. ☐ None of these.
2. ☐ Compilation Error.
3. ☐ It will print "Hello Java" .... 20 times for obj1 and obj2.
4. ☐ DeadLock.

**Incorrect**

codewitharrays.in 8007592194