1: Mr. John is working in XYZ Company Ltd. He tries to compile and execute the following program:

```
class SCJPQ10 {
 public static void main(String args[]) {
 SCJPQ10 scjp = new SCJPQ10();
 scjp.method1();
 }
 public void method1() {
 ThreadAsc tasc = new ThreadAsc("OneThread");
 tasc.start();
 }
}
class ThreadAsc extends Thread {
 private String str1 = " ";
 ThreadAsc(String s) {
 str1 = s;
 }
 public void run() {
 methodWait();
 System.out.println("Thread Completed");
 }
 public void methodWait() {
 while (true) {
 try {
 System.out.println("Waiting Thread");
 wait();
 } catch (InterruptedException e) {
 }
 System.out.println(str1);
 }
 }
}
```

What will happen when he compiles and executes the preceding program?

A. The program displays "Waiting Thread" as an output.

B. The program displays "Thread Completed" as an output.

C. The program compiles successfully but an exception throws at runtime after displaying “Waiting Thread”.

D. The program generates compile-time error.

E. The program displays "Waiting Thread" followed by "Thread Completed" as an output.

Q 2: Maria is working as a Java Programmer in XYZ Solutions. She tries to compile and execute the following program:

```
class SCJPQ20 implements Runnable {
 int k = 0;
 public SCJPQ20(int i) {
 this.k = i;
 }
 public static void main(String[] args)
 {
 new SCJPQ20(2).run();
 new SCJPQ20(1).run();
 }
 public void run() {
 for(int i=0; i<k; i++) {
 System.out.println("run() method...");
 }
 }
}
```

What will happen when she compiles and executes the preceding program?
A. The program displays ″run() method...″ two times.
B. The program displays ″run() method...″ as an output.
C. The program displays ″run() method...″ three times.
D. The program creates two new threads.

Q 3: Assume that you are a Software Engineer and attempts to compile and execute the following program:

```
class SCJPQ25 extends Thread {
 SCJPQ25() {
 setPriority(5);
 }
 public void run() {
 System.out.println("Thread running");
 }
 public static void main(String args[]) {
 SCJPQ25 th1 = new SCJPQ25();
 SCJPQ25 th2 = new SCJPQ25();
 SCJPQ25 th3 = new SCJPQ25();
 th1.start();
 th2.start();
 th3.start();
 }
}
```

Which of the following statements are true about the preceding program?
A. When the program runs, all three threads (th1, th2, and th3) executes concurrently, taking time-sliced
turns in the CPU.
B. The thread th1, and th2 executes but th3 never get the CPU.
C. When the program runs, thread th1 executes first, then th2 executes and then th3 executes.
D. None of the above options are true.

Q 4: Ramesh is working as a Java Developer in ABC Software Company Ltd. He tries to compile and execute
the following program:

```
class ThreadTest extends Thread {
 String str = "";
 public ThreadTest(String s) {
 this.str = s;
 }
 public void run() {
 if(str.equals("thread1")) {
 yield();
 }
 System.out.println("End of " + str);
 } public static void main(String args []) {
 Thread thread1 = new ThreadTest("thread1");
 thread1.setPriority(Thread.MAX_PRIORITY);
 Thread thread2 = new ThreadTest("thread2");
 thread2.setPriority(Thread.MIN_PRIORITY);
 thread1.start(); thread2.start();
 }
}
```

What will happen when he compiles and executes the preceding program?

A. The program displays ″End of thread1″ followed by ″End of thread2″ as an output.

B. The program displays ″End of thread1″ and ″End of thread2″ in random order.

C. The program generates exception at runtime.

D. The program generates error at compile-time.

Q 5: Imagine that you are a Java Developer and attempts to compile and execute the following program:

```
class MyThread extends Thread {
 public void run() {
 System.out.println("Executing while loop");
 while(true){}
 }
} public class ThreadTest {
 public static void main(String args[]) throws Exception {
 MyThread thread1 = new MyThread();
 thread1.start();
 Thread.sleep(5000);
 thread1.interrupt();
 }
}
```

Which of the following statements are correct about the preceding program?

A. The program executes and will never end. B. The program generates compiler-time error.
C. The program generates runtime exception. D. None of the above options are correct.

Q 6: Which of the following methods are used to pass a timeout argument?

A. wait() B. yield() C. start() D. join()
E. notify() F. notifyAll() G. run() H. sleep()

Q 7: Imagine that you are a Java Developer and attempts to compile and execute the following program:

```
class DeadlockTest
{
 static StringBuffer s1 = new StringBuffer();
 static StringBuffer s2 = new StringBuffer();
 public static void main(String[] args) {
 new Thread (
 new Runnable() {
 public void run() {
 synchronized(s1) {
 s1.append("A");
 synchronized(s2) {
 s2.append("B");
 }
 }
 System.out.println(s1);
 }
 }
 ).start();
 new Thread
 (
 new Runnable() {
 public void run() {
 synchronized(s2) {
 s1.append("B");
 synchronized(s1) {
 s2.append("A");
 }
 }
 System.out.println(s2);
 }
 }
 ).start();
 }
}
```

What will happen when you compile and execute the preceding program?

A. Program result in dead lock situation.

B. Program displays "AA" followed by "BB" as an output.

C. Program displays "BB" followed by "AA" as an output.

D. Program displays "AB" followed by "BA" as an output.

Q 8: Assume that you are a Java Developer and attempt to compile and execute the following program:

```
class MyThread extends Thread {
 public void run() {
 try {
 sleep(5000);
 } catch (InterruptedException e) {
 System.out.println("Exception " + e);
 }
 }
 public static void main(String args[]) {
 MyThread thread1 = new MyThread();
 long startTime = System.currentTimeMillis();
 thread1.start();
 System.out.print("Time required to execute thread1 is "
 + (System.currentTimeMillis() - startTime));
 }
}
```

What will happen when you compile and execute the preceding program?

A. Program displays "5000" as an output.

B. Program displays a number greater than or equal to 0.

C. Program displays a number greater than 5000.

D. Program generates compile-time error.

Q 9: Assume that you are a Software Developer and attempt to compile and execute the following program:

```
class X implements Runnable {
 public void run() {
 }
}
class Y {
 public static void main(String args[]) {
 Thread th1 = new Thread(); //1
 Thread th2 = new Thread(new X()); //2
 Thread th3 = new Thread(new X(), "X"); //3
 Thread th4 = new Thread("X"); //4
 }
}
```

At which line is a compile-time error generated?
A. Statement at //1 B. Statement at //2
C. Statement at //3 D. None of the above

Q 10: Assume that you are a Java Developer in XYZ Software Solution and tries to compile and run the following thread program:

```
class ThreadClass implements Runnable {
 public void run() {
 System.out.println("Running thread...");
 }
}
class Test {
 public static void main(String args[]) throws Exception {
 Thread thread1 = new Thread();
 Thread thread2 = new Thread(new ThreadClass());
 Thread thread3 = new Thread(new ThreadClass(), "Thread3");
 Thread thread4 = new Thread("Thread4");
 Thread thread5 = new Thread("Thread5", 5); //1
 Thread thread6 = new Thread("Thread6", new MyClass()); //2
 }
}
```

What happen when you compile and run the preceding program?
A. Program generates compilation error at //1
B. Program generates compilation error at //2
C. Program generates runtime error
D. Program compiles successfully and displays Running thread...

Q 11: Jude during her training session was asked to create a program extending the Thread class to implement threading concept. Jude created the following program:

```
class Jude extends Thread {
 public void run() {
 for(int i=0;i<2;i++) {
 System.out.print("Hello"+” “);
 }
 txt("Java");
 }
 public void txt(String str) {
 String s=str;
 System.out.print(s);
 }
 public static void main(String args[]) {
 Jude thrd=new Jude();
 thrd.start();
 }
}
```

What will be the output of this program?
A. The program will generate compile time error
B. The program will display Java Hello Hello as output.
C. The program will display Hello Hello Java as output.
D. The program will display Hello Hello as output.

Q 12: Jane and Sam, while preparing for the SCJP exam, came across the following program:

```
class Jude implements Runnable {
 static Thread t;
 public void run() {
 try{
 for(int i=0;i<3-1;i=i+2, i--) {
 System.out.println("Hello");
 t.sleep(1500);
 }
 }catch(InterruptedException ie) {
 ie.printStackTrace();
 }
 }
 public static void main(String ar[]) {
 Jude jd=new Jude();
 t = new Thread(jd);
 t.start();
 }
}
```

What will be the output of this program?

A. The program displays Hello followed by Hello with a pause of 1500 miliseconds.

B. The program displays Hello Hello Hello.

C. The program will compilation error.

D. Program will successfully compile and execute without displaying any value.

Q 13: Rose works as a developer in XYZ Company and she created the following program:

```
class Test implements Runnable {
 static Thread t1, t2;
 public void run() {
 for(int i=0; ;){
 System.out.println(Thread.currentThread().getName());
 i++;
 }
 }
 public static void main(String ar[]) {
 Test t=new Test();
 t1=new Thread(t, "T1");
 t2=new Thread(t, "T2");
 try{ t1.start();
 t1.sleep(5000);
 }catch(Exception e){ }
 t2.start();
 }
}
```

What will be the output of this program?

A. The program will cause compilation error.

B. Program will execute in an infinite loop and display T1 and T2 depending on their time slice.

C. T2 will not be displayed because the t1.start() first invokes the run() method and therefore T2 will not be
displayed

D. The program will throw runtime exception.

Q 14: Jude works as a developer in XYZ Company and during a project she created the following program:

```
class Test implements Runnable {
 static Thread t1;
 public void run() {
 for(int i=0; i<2;i++){
 System.out.print("Hello"+” “);
 }
 }
 public static void main(String args[]) {
 Test t=new Test();
 t1=new Thread(t);
 t1.run();
 }
}
```

What will be the output of the preceding program?

A. The program will display Hello Hello as output.

B. Program will not compile successfully because run() method is being explicitly called.

C. The run() method that will execute in this program, is of Runnable interface.

D. Program will successfully compile and execute but does not display any value.

Q 15: Rose while working on a project in XYZ Company created the following program:

```
class Test implements Runnable {
 static Thread t1, t2;
 public void run() {
 synchronized(this){
 try{
 for(int i=1; i<=5;i++) {
 System.out.println(i);
 t2.wait();
}
 }catch(Exception e) { }
 }
 }
public static void main(String ar[])
 {
 Test t=new Test();
 t1=new Thread(t);
 t2=new Thread(t);
 t1.start();
 t1.start();
 }
}
```

What will be the output of the preceding program?

A. The program generates compilation error because the start() method is called twice on the same object.

B. The program throws runtime exception because the start() method is called twice on the same object.

C. The program displays 1 followed by runtime exception.

D. The program displays 1 2 3 4 5 as output.

Q 16: Rose during her training session was shown the following program:

```
class Rose implements Runnable {
 public void run() {
 System.out.println(" Hello");
 }
 public static void main(String ar[]) {
 Rose thrd=new Rose ();
 Thread T=new Thread(thrd);
 T.run();
 }
}
```

What will be the output of the preceding program?

A. The program will display Hello as output.

B. The program will generate compilation error because start() method is not invoked therefore run()
method can also not be invoked.

C. The program will throw runtime exception because start() method is not invoked therefore run() method
can also not be invoked.

D. Program will compile and execute successfully but nothing will be displayed as output.

Q 17: Assume that you are a Java Developer in ABC Software Solution and tries to compile and run the following thread program:

```
class ThreadTest extends Thread {
 public void run() {
 System.out.println("Starting thread");
 try {
 Thread.sleep(1000);
 System.out.println("Time is up");
 } catch (InterruptedException ex) {
 System.out.println("Interrupted" +ex);
 }
 }
}
class Test1
{
 public static void main(String args[])
 {
 ThreadTest t1 = new ThreadTest();
 ThreadTest t2 = new ThreadTest();
 t1.start();
 t2.start();
 }
}
```

What happen when you compile and run the preceding program?

A. Program displays "Starting thread” 2 times and after 1 seconds, it displays "Time is up” 2 times.

B. Program does not display any output

C. Program generates compilation error

D. Program generates runtime error.

Q 18: Imagine you are working in ABC Company as a Java programmer and have written the following program:

```
public class Concur1 {
 public static void main(String ar[]) {
 MyThread mthread = new MyThread();
 mthread.start();
 }
}
class ThreadDemo implements Runnable {
 public void run() {
 System.out.println("Thread Demo");
 }
}
class MyThread extends Thread {
 public void start(){
 System.out.println("My Thread");
 }
 public void run() {
 Thread thrd = new Thread(new ThreadDemo());
 thrd.start();
 }
}
```

What is the result when you compile and execute this program?

A. The program compiles without error and displays My Thread as output.

B. The program compiles without error and displays My Thread and Thread Demo as output.

C. The program compiles without error and displays Thread Demo as output.

D. The program generates compilation error because of overridden start() method.

Q 19: Imagine you are working as a Java programmer in ABC Company and have written the following program:

```
public class Concur2 {
 public static void main(String ar[]) {
 Thread inactthrd = new Thread(new ThreadInactiveDemo());
 inactthrd.start();
 Thread actthrd = new Thread(new ThreadActiveDemo());
 actthrd.start();
 }
}
class ThreadActiveDemo implements Runnable {
 public void run() {
 System.out.println("Thread - Active Demo");
 }
}
class ThreadInactiveDemo implements Runnable {
 public void run() {
Question Bank 4: Concurrency 11
 try {
 Thread.sleep(4 * 1000);
 } catch (InterruptedException E) {
 E.printStackTrace();
 }
 System.out.println("Thread - Inactive Demo");
 }
}
```

What will be the result when you compile and execute this program?

A. The program compiles without error and displays Thread – Inactive Demo, takes pause and then
displays Thread – Active Demo.

B. The program generates compilation error.

C. The program compiles without error and displays Thread – Active Demo, takes pause and then displays
Thread – Inactive Demo.

D. Different output will be shown in multiple executions.

Q 20: Imagine you are a Java programmer in ABC Company and you have written the following code

```
public class Concur3 {
 public static void main(String ar[]) {
 MyThread thrd = new MyThread();
 thrd.start();
 }
}
class MyThread extends Thread {
 public void run() {
 for (int var=1; var<6 ; var++) {
 if (Thread.interrupted()) {
 System.out.println("The interrupt() Method Invoked");
 }
 if (var == 3 || var == 5) {
 this.interrupt();
 }
 }
 }
}
```

What is the result when you compile and execute this program?

A. It shows a compile time error.

B. It throws InterruptedException at runtime.

C. It compiles without error and does not display any output.

D. It compiles without error and displays The interrupt() Method Invoked as output.

Q 21: Imagine you are a Java programmer of ABC Company and have written the following code

```
public class Concur6 {
 public static void main(String ar[]) throws InterruptedException {
 System.out.println("The main() method");
12 SCJP: Sun Certified Programmer for Java 6
 DaemonThreadDemo thrd = new DaemonThreadDemo();
 thrd.start();
 thrd.join();
 System.out.println(thrd.isAlive());
 }
}
class DaemonThreadDemo extends Thread {
 public DaemonThreadDemo() {
 setDaemon(true);
 }
 public void run() {
 System.out.println("The run() method");
 }
}
```

What is the result of the program when you compile and execute it?

A. The program displays The run() method, The main() method and false as output.

B. The program displays The run() method, The main() method and true as output.

C. The program displays The main() method, The run() method and false as output.

D. The program displays The main() method, The run() method and true as output.

Q 22: Imagine you are working in ABC Company as Java programmer and have written following code:

```
public class Concur7 {
 public static void main(String ar[]) throws InterruptedException {
 String str1 = new String("R1");
 String str2 = new String("R2");
 MyThread thrd1 = new MyThread("Firsr", str1, str2);
 MyThread thrd2 = new MyThread("Second", str1, str2);
 thrd1.start();
 thrd1.join();
 thrd2.start();
 }
}
class MyThread extends Thread {
 private String First;
 private String Second;
 public MyThread(String thrdname, String one, String two) {
 super(thrdname);
 First = one;
 Second = two;
 }
 public void run() {
 if (getName().equals("First")) {
 synchronized (First) {
 try {
 Thread.sleep(4000);
 } catch (InterruptedException E) {
 E.printStackTrace();
 }
 synchronized (Second) {}
 }
 }
 else {
 synchronized (Second) {
 try {
 Thread.sleep(4000);
 } catch (InterruptedException E) {
 E.printStackTrace();
 }
 synchronized (First) {}
 }
 }
 }
}
```

What will be the result when you compile and execute the preceding code?

A. The program generates compilation error.
B. The program compiles without error but throws runtime exception.
C. The program compiles without error but deadlock occurs at runtime.
D. The program compiles and executes without any error or deadlock

Q 23: Imagine as a Java programmer you have written the following code to implement the wait() and notifyAll() methods.

```
public class Concur8 {
 public static void main(String ar[]) throws InterruptedException {
 Thread thrd2 = new Thread("thrd2");
 synchronized (Concur8.class) {
 thrd2.wait();
 }
 Thread thrd1 = new Thread("thrd1");
 synchronized (thrd1) {
 thrd1.wait();
 thrd1.notifyAll();
 }
 }
}
```

What is the result when you compile and execute the program?
A. The program compiles without error but throws IllegalMonitorStateException exception on thread t2 at runtime.
B. The program generates compilation error.
C. The program compiles without error and executes in an infinite loop.
D. The program compiles without error but throws InterruptedException exception at runtime.