***Multi-Threading***

**Q: 07 Which two code fragments will execute the method doStuff() in a separate**

**thread? (Choose two.)**

A. new Thread() {

public void run() { doStuff(); }

};

B. new Thread() {

public void start() { doStuff(); }

};

C. new Thread() {

public void start() { doStuff(); }

}.run();

D. new Thread() {

public void run() { doStuff(); }

}.start();

E. new Thread(new Runnable() {

public void run() { doStuff(); }

}).run();

F. new Thread(new Runnable() {

public void run() { doStuff(); }

}).start();

**Answer: D, F**

**Q: 08 Given:**

**1. public class TestOne implements Runnable {**

**2. public static void main (String[] args) throws Exception {**

**3. Thread t = new Thread(new TestOne());**

**4. t.start();**

**5. System.out.print("Started");**

**6. t.join();**

**7. System.out.print("Complete");**

**8. }**

**9. public void run() {**

**10. for (int i = 0; i < 4; i++) {**

**11. System.out.print(i);**

**12. }**

**13. }**

**14. }**

**What can be a result?**

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes and prints "StartedComplete".

D. The code executes and prints "StartedComplete0123".

E. The code executes and prints "Started0123Complete".

**Answer: E**

**Q: 09Given:**

**1. public class TestOne {**

**2. public static void main (String[] args) throws Exception {**

**3. Thread.sleep(3000);**

**4. System.out.println("sleep");**

**5. }**

**6. }**

**What is the result?**

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes normally and prints "sleep".

D. The code executes normally, but nothing is printed.

**Answer: C**

**Q: 10 Given:**

**11. public class Test {**

**12. public enum Dogs {collie, harrier, shepherd};**

**13. public static void main(String [] args) {**

**14. Dogs myDog = Dogs.shepherd;**

**15. switch (myDog) {**

**16. case collie:**

**17. System.out.print("collie ");**

**18. case default:**

**19. System.out.print("retriever ");**

**20. case harrier:**

**21. System.out.print("harrier ");**

**22. }**

**23. }**

**24. }**

**What is the result?**

A. harrier

B. shepherd

C. retriever

D. Compilation fails.

E. retriever harrier

F. An exception is thrown at runtime.

**Answer: D**

**Q: 11 Given:**

**11. Runnable r = new Runnable() {**

**12. public void run() {**

**13. System.out.print("Cat");**

**14. }**

**15. };**

**16. Thread t = new Thread(r) {**

**17. public void run() {**

**18. System.out.print("Dog");**

**19. }**

**20. };**

**21. t.start();**

**What is the result?**

A. Cat

B. Dog

C. Compilation fails.

D. The code runs with no output.

E. An exception is thrown at runtime.

**Answer: B**

**Q: 17 Given:**

**foo and bar are public references available to many other threads. foo refers to a Thread and bar is an**

**Object. The thread foo is currently executing bar.wait().**

**From another thread, what provides the most reliable way to ensure that foo will stop executing wait()?**

A. foo.notify(); B. bar.notify();

C. foo.notifyAll(); D. Thread.notify();

E. bar.notifyAll(); F. Object.notify();

**Answer: E**

**Q: 18 Given:**

**1. public class MyLogger {**

**2. private StringBuilder logger = new StringBuuilder();**

**3. public void log(String message, String user) {**

**4. logger.append(message);**

**5. logger.append(user);**

**6. }**

**7. }**

**The programmer must guarantee that a single MyLogger object works properly for a multi-threaded system.**

**How must this code be changed to be thread-safe?**

A. synchronize the log method

B. replace StringBuilder with StringBuffer

C. replace StringBuilder with just a String object and use the string concatenation (+=) within the log method

D. No change is necessary, the current MyLogger code is already thread-safe.

**Answer: A**

**Q: 19 Given:**

**1. public class TestSeven extends Thread {**

**2. private static int x;**

**3. public synchronized void doThings() {**

**4. int current = x;**

**5. current++;**

**6. x = current;**

**7. }**

**8. public void run() {**

**9. doThings();**

**10. }**

**11.}**

**Which statement is true?**

A. Compilation fails.

B. An exception is thrown at runtime.

C. Synchronizing the run() method would make the class thread-safe.

D. The data in variable "x" are protected from concurrent access problems.

E. Declaring the doThings() method as static would make the class thread-safe.

F. Wrapping the statements within doThings() in a synchronized(new Object()) { } block would make the class thread-safe.

**Answer: E**

**Q: 21 Given:**

**1. public class Threads5 {**

**2. public static void main (String[] args) {**

**3. new Thread(new Runnable() {**

**4. public void run() {**

**5. System.out.print("bar");**

**6. }}).start();**

**7. }**

**8. }**

**What is the result?**

A. Compilation fails.

B. An exception is thrown at runtime.

C. The code executes normally and prints "bar".

D. The code executes normally, but nothing prints.

**Answer: C**

**Q: 22 Which three will compile and run without exception? (Choose three.)**

A. private synchronized Object o;

B. void go() {

synchronized() { /\* code here \*/ }

C. public synchronized void go() { /\* code here \*/ }

D. private synchronized(this) void go() { /\* code here \*/ }

E. void go() {

synchronized(Object.class) { /\* code here \*/ }

F. void go() {

Object o = new Object();

synchronized(o) { /\* code here \*/ }

**Answer: C, E, F**