* **Question 1**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | The notify() method non-deterministically chooses one waiting thread to de-queue |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 2**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | A race condition can be avoided by using synchronised methods |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 3**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Incorrect | Thread priorities can be used to ensure one thread is executed in preference to another |  |  |  |
| |  |  | | --- | --- | | Answers: | True | |  | Correct False | |  |  |  |

* **Question 4**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | The run() method is a new thread's equivalent of an applications main() method |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 5**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | Therac-25 code was not designed for clean automated testing |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 6**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Incorrect | If a running thread is cancelled, its execution is terminated as soon as possible |  |  |  |
| |  |  | | --- | --- | | Answers: | True | |  | Correct False | |  |  |  |

* **Question 7**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | An object is thread safe if its methods can be called safely by multiple threads |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 8**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | A multi-threaded application can avoid deadlock by calling notifyAll() rather than notify() |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 9**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | The run() method can be defined using a Runnable lambda expression |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 10**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | A deadlock is caused by a cycle of threads each waiting on the next |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 11**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | The run() method can be defined by overriding the default Thread implementation |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 12**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | A blocking remove will wait if necessary for another thread to queue an item |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 13**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | Java.util.concurrent provides a thread-safe concurrent hash map |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 14**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | An update anomaly can occur because thread execution is interleaved |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 15**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Incorrect | A Future<T> object is used by a worker thread to stream results of type T |  |  |  |
| |  |  | | --- | --- | | Answers: | True | |  | Correct False | |  |  |  |

* **Question 16**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Incorrect | Therac-25 failed because it replaced hardware interlocks with software checks |  |  |  |
| |  |  | | --- | --- | | Answers: | True | |  | Correct False | |  |  |  |

* **Question 17**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Incorrect | The sleep(n) method ensures a thread does not execute for at least n millisecs |  |  |  |
| |  |  | | --- | --- | | Answers: | True | |  | Correct False | |  |  |  |

* **Question 18**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | Therac-25 code synchronisation failures led to race conditions |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 19**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | Each Java object has an associated lock and wait queue |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 20**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | Multi-tasking involves multiple parallel processes that do not share memory |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 21**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | If there is a race condition, the losing thread's update decides the final value |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 22**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Incorrect | Therac-25 coding errors were not the primary reason for the injuries & deaths it caused |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 23**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | Two invocations of a synchronised method must occur at the same time |  |  |  |
| |  |  | | --- | --- | | Answers: | True | |  | Correct False | |  |  |  |

* **Question 24**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | Some JavaFX methods need not be invoked from the JavaFX Application thread |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 25**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Incorrect | Switching between threads is often O/S specific and non-portable |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 26**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Incorrect | Multi-threading involves multiple parallel threads that share memory |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 27**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | Earlier languages like C and C++ did not have built-in support for threads |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 28**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | Java thread pools can be dynamic in size or limited to the number of CPU cores |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 29**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Incorrect | Java threading behaviour is non-deterministic and not portable |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 30**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | When a JavaFX worker thread succeeds, the JavaFX Application thread called the onSucceeded() event handler |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 31**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | An update anomaly can occur even if only one thread writes to shared data at any given time |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 32**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Incorrect | It is not possible to support threading in a portable deterministic way |  |  |  |
| |  |  | | --- | --- | | Answers: | True | |  | Correct False | |  |  |  |

* **Question 33**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | Update anomalies can be avoided by only sharing volatile primitive variables |  |  |  |
| |  |  | | --- | --- | | Answers: | True | |  | Correct False | |  |  |  |

* **Question 34**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | JavaFX background threads define a call() method for their processing logic |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 35**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | JavaFX has support for background task progress bars |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 36**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Incorrect | JavaFX animations should be programmed using a separate thread and sleep() calls |  |  |  |
| |  |  | | --- | --- | | Answers: | True | |  | Correct False | |  |  |  |

* **Question 37**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | An update anomaly can occur if two threads execute their code simultaneously |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |

* **Question 38**

0 out of 0 points

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | | |
| Correct | Threading code must make trade offs between safety and performance |  |  |  |
| |  |  | | --- | --- | | Answers: | Correct True | |  | False | |  |  |  |