

Programming with Haiku

Unit 1 Review

Written by DarkWorm



Lesson 1

1. What are templates?
2. What are templates most often used to create?
3. How is a template-based class declaration different from a regular one? A template-based function declaration?
4. Why must functions which use templates be defined in a header file?
5. What is the difference between a vector and a deque?
6. What is the main drawback of the list container?
7. What is the namespace used by the Standard Template Library?
8. What does the using keyword do?

Lesson 2

1. What is the difference between the map and set containers?
2. What is the purpose of the equal_range() method used by multimap and multiset?
3. What is FIFO processing and which container adapter is well-suited to it? LIFO processing?
4. What is the difference between the queue and priority_queue containers?

Lesson 3

1. What are the four standard C++ streams?
2. What is the difference between istream::bad() and istream()::fail()?
3. Why should the endl constant be used instead of '\n' when writing an end-of-line character to a C++ stream?
4. What is the call stack?
5. What does the boolalpha formatter do?
6. Why are exceptions not normally used in Haiku?

Lesson 4

1. Why was the Subversion project started?
2. What setup work should be done before Mercurial is used?
3. What is the purpose of the shell variable EDITOR in relation to Mercurial?
4. What command creates a Mercurial repository?
5. What is the difference between hg commit and hg push?
6. What does hg revert do?