Directions to Candidates:
Time Allowed: 3 hours 15 minutes
Number of Questions: 7 questions
The exam is worth 60% of the total marks for the subject.
Questions 1 and 2 are worth 10 marks each
Questions 3 to 7 are worth 8 marks each and contain sub-questions worth 1-3mks each

Question 1: (Multiple Choice Questions)
Question 2: (True-False questions)
Question 3: (Sorting and Searching)
Question 4: (Pointers and Dynamic Memory)
Question 5: (Classes, Templates and Inheritance)
Question 6: (Collection Classes)
Question 7: (Trees and Heaps)

Exam topics:
Here are some topics covered by the exam:

- sorting algorithms
- quicksort
- pointers
- classes
- structs
- public
- private
- inheritance
- linked list
- binary tree
- hashing
- binary search
- pointers
- dynamic memory
- destructor
- friend function and class
- virtual
- heap
- templates
- delete
- new
- sizeof()
Exam Tips

You will find the answers to some of the revision questions here: http://www.uow.edu.au/~koren/csci124/

There are two older csci121 exams from the library there too.

Here are some tips on the exam:

Most of the questions in this exam are not questions requiring you to write any C++ code. Most just test your understanding of C++ and other topics covered in the course. For example, the M/C & T/F questions are worth 20 marks alone. About half of the sub-questions in Questions 3 to 7 are just short answer questions, not coding questions. All the sub-questions of Questions 3 to 7 are worth 1, 2 or 3 marks each only. Some questions are on sorting algorithms, trees, lists, hashing, etc. with no code. Many have small code fragments and ask you something about it.

So you should be able to write the answers to half of the exam questions straight down without writing any code if you understand the material in the course notes. However, there are one or two questions which are a bit tricky on C++ code. My best advice for preparing for these questions is to read the notes in detail after looking at the revision questions and past exam questions.

Most of the C++ coding questions do not require much code to be written. I have noticed that the past exams tend to have longer coding questions and more of them than in this exam. This exam is tending to test if you did the assignment coding and can recall what you did. Practise writing some of the smaller functions in the assignments that occur often including some of the functions on lists, trees, etc. Use the code in the notes here also. Practise question on pointers.

There are no complete programs to write with templates, inheritance or overloaded operators. But there is a question or two on converting some code to templates and overloaded operators. There is a derived class with inherited functions and data that you have to write a function for. (This advanced coding is not worth many marks.)

Here is a good strategy if you need to build up your confidence for the exam. Start by working out how many marks you need from the exam to pass the subject. A pass conceded (45%) is also worth 6 credit points toward your degree. When you sit for the exam, first look at the questions and mark all the questions you know you can do. You might circle any M/C and T/F questions that you know the answer too. Also, quickly, write on the exam paper the answer to any short answer questions you know. Roughly sketch any code you know. Spend about the first 15-20 minutes on this. Just focus on getting the marks to get over the line. This helps you to build your confidence. When you know you have got what you need to pass the subject, write the answers in the answer booklet and then set your mind on getting enough additional marks for a credit, distinction and high-distinction pass, in that order. Do not leave any M/C or T/F questions unanswered. Guess any answers you don't know. Also, there may be code in the exam paper that can help to answer previous questions. So if you can’t remember how to write certain code, browse through the paper to see if any of the code in the exam reveals or reminds you of the correct answer. Always, look at how many more questions you have to do after you complete each question and check the time. You don’t want to run out of time and leave any question unfinished that you can do.