UTS

SEAT NUMBER:

STUDENT NUMBER:

SURNAME:
(FAMILY NAME)

OTHER NAMES:

This paper and all materials issued must be returned at the end of the examination. They are not to be removed from the exam centre.

Examination Conditions:

It is your responsibility to fill out and complete your details in the space provided on all the examination material provided to you. Use the time before your examination to do so as you will not be allowed any extra time once the exam has ended.

You are **not** permitted to have on your desk or on your person any unauthorised material. This includes but not limited to:

- Mobile phones
- Smart watches and bands
- Electronic devices
- Draft paper (unless provided)
- Textbooks (unless specified)
- Notes (unless specified)

You are **not** permitted to obtain assistance by improper means or ask for help from or give help to any other person.

If you wish to **leave and be re-admitted** (including to use the toilet), you have to wait until **90 mins** has elapsed.

If you wish to **leave the exam room permanently**, you have to wait until **60 mins** has elapsed.

You are not permitted to leave your seat (including to use the toilet) during the final 15 mins.

During the examination **you must first seek permission** (by raising your hand) from a supervisor before:

- Leaving early
- Using the toilet
- Accessing your bag

Misconduct action will be taken against you if you breach university rules.

Declaration: I declare that I have read the advice above on examination conduct and listened to the examination supervisor's instructions for this exam. In addition, I am aware of the university's rules regarding misconduct during examinations. I am not in possession of, nor do I have access to, any unauthorised material during this examination. I agree to be bound by the university's rules, codes of conduct, and other policies relating to examinations.

Signature:

Date:

31251 Data Structures and Algorithms

Time Allowed: 120 minutes.

Reading time: 10 minutes.

Reading time is for reading only. You are not permitted to write, calculate or mark your paper in any way during reading time.

Open Book

Permitted materials for this exam:

Dictionary
Drawing Instruments
Lecture Notes
Notes
Textbook

Materials provided for this exam:

1 x 20 Page Booklet
1 x General Purpose Answer Sheets (GPAS-240R)

Students please note:

All code fragments are in C++, using the C++11 standard (as used in the subject).

Do not open your exam paper until instructed.

Rough work space
Do **not** write your answers on this page.

Question 1 (1 mark)

What will the following code snippet print out?

```
int i = 5;
int * p = &i;
*p += 10;
cout << p;
```

A. The address of variable `i`.
B. 15.
C. 10.
D. An unknown value from memory.

Question 2 (1 mark)

Which of the following code fragments correctly dereferences a pointer called `p`?

A. `*p`
B. `&p`
C. `p*`
D. `+p`

Question 3 (1 mark)

Which of the following code fragments gives the address of an `int` variable called `i`?

A. `*i`
B. `&i`
C. `i->`
D. `i*`

Question 4 (1 mark)

What will the following code snippet print out?

```
int i = 5;
int * p = &i;
*p += 10;
cout << *p;
```

A. The address of variable `i`.
B. 15.
C. 10.
D. An unknown value from memory.

Question 5 (1 mark)

Consider the following class:

```
class node {
      private:
            int data;
            node* next;
      public:
            node(int data, node* next){
                  this->data = data;
                  this->next = next;
            }

            int get_data() const { return this->data; }

            bool p() const {
                  if (next == nullptr) return true;
                  if (data < next->get_data()) return false;
                  return next->p();
            }
};
```

and the following code snippet:

```
node* x = new node(1, nullptr);
node* head = new node(2, x);
head->p();
```

What is the result of the code snippet?

A. `p` is executed on both `node`s and both return `false`.
B. `p` is executed on both `node`s and both return `true`.
C. `p` is executed on only the first node and it returns `false`.
D. `p` is executed on only the first node and it returns `true`.

Question 6 (1 mark)

Given a stack with the standard push and pop functions, what value is on the top of the stack after the following sequence of operations: `push(5), push(10), push(9), pop(), push(1), push(1), pop(), push(6), pop(), pop()`?

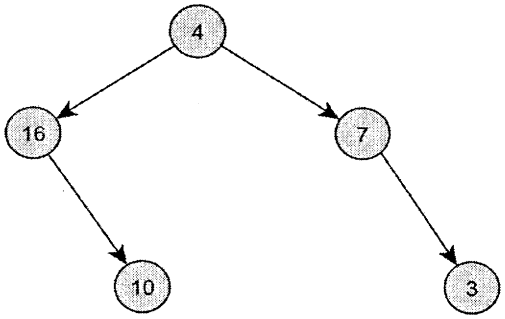
A. 5
B. 9
C. 10
D. 1

Question 7 (1 mark)

Given a standard linked list, what is the asymptotic (big-oh) time for inserting an element into the middle of a list of length n?

A. $O(1)$
B. $O(\log_2 n)$
C. $O(n)$
D. $O(n^2)$

Questions 8-11 refer to the following tree:



Question 8 (1 mark)

What is the visiting order of a depth-first preorder traversal of the tree?

A. 4, 16, 7, 10, 3
B. 4, 16, 10, 7, 3
C. 16, 10, 4, 7, 3
D. 10, 16, 3, 7, 4

Question 9 (1 mark)

What is the visiting order of a depth-first inorder traversal of the tree?

A. 4, 16, 7, 10, 3
B. 4, 16, 10, 7, 3
C. 16, 10, 4, 7, 3
D. 10, 16, 3, 7, 4

Question 10 (1 mark)

What is the visiting order of a depth-first postorder traversal of the tree?

A. 4, 16, 7, 10, 3
B. 4, 16, 10, 7, 3
C. 16, 10, 4, 7, 3
D. 10, 16, 3, 7, 4

Question 11 (1 mark)

What is the visiting order of a breadth-first traversal of the tree?

A. 4, 16, 7, 10, 3
B. 4, 16, 10, 7, 3
C. 16, 10, 4, 7, 3
D. 10, 16, 3, 7, 4

Question 12 (1 mark)

Which one of the following statements is TRUE?

A. $15n^2 + 3n + 2 \in O(n \log_2 n)$
B. $15n^2 + 3n + 2 \in \Omega(n^4)$
C. $15n^2 + 3n + 2 \in O(n^3)$
D. $15n^2 + 3n + 2 \in O(15)$

Question 13 (1 mark)

If a mathematical function g is at least as big as another mathematical function f for every value greater than 10, which of the following is necessarily TRUE?

A. $g \in O(f)$
B. $f \in O(g)$
C. $f \in \Omega(g)$
D. $f \in \Theta(g)$

Question 14 (1 mark)

Which one of the following statements is FALSE?

A. $n^5 + 2n^4 - 2n + 2 \in O(n^6)$
B. $n^5 + 2n^4 - 2n + 2 \in \Omega(n^4)$
C. $n^5 + 2n^4 - 2n + 2 \in \Theta(n^5)$
D. $n^5 + 2n^4 - 2n + 2 \in \Theta(n^7)$

Question 15 (1 mark)

Which of the following operations is NOT normally included in a Map ADT?

A. add(K key, V value)
B. get(K key)
C. pop()
D. contains(K key)

Question 16 (1 mark)

Which of the following operations is NOT normally included in a List ADT?

A. void enqueue(V value)
B. void prepend(V value)
C. void append(V value)
D. bool isEmpty()

Question 17 (1 mark)

Assuming there is no collision, what is the best expected asymptotic (big-oh) running time of inserting a new element into a hashmap which already has *n* elements?

A. $O(1)$
B. $O(n)$
C. $O(n^2)$
D. $O(\log_2 n)$

Question 18 (1 mark)

Which algorithmic paradigm does the Rabin-Karp string searching algorithm best match?

A. Greedy Algorithms.
B. Dynamic Programming.
C. Divide and Conquer.
D. Binary Search.

Question 19 (1 mark)

Which algorithmic paradigm does Mergesort best match?

A. Greedy Algorithms.
B. Dynamic Programming.
C. Divide and Conquer.
D. Binary Search.

Question 20 (1 mark)

Which of the following statements is FALSE?

A. Every recursive algorithm can be rewritten as an iterative algorithm.
B. Every iterative algorithm can be rewritten as a recursive algorithm.
C. Iteration is always better than recursion.
D. Recursion requires the explicit or implicit use of a stack.

Part B: Short Answer Questions

Question 21 (10 marks)

This question concerns hashmaps, hash functions, their properties and their implementation.

i. (2 marks) What is a hash function?
ii. (2 marks) List two desirable properties of a good hash function.
iii. (2 marks) What is a collision in the context of a hash function?
iv. (4 marks) Begin with an empty hashmap of size 5 (implemented using an array) which only stores integer data. Using the data as the key, insert one by one the entries 6, 2, 7, 13 and 1 in this order into the hashmap using linear probing to resolve collisions and hash function h(K) = K mod N. Show the map after each insertion and indicate how you computed the insertion point.

Question 22 (10 marks)

This question is a short answer *"explain in plain English"* question. You will be given some code and you are required to describe the purpose of it. For example, given the following code:

```
if (a < b)
      std::cout << a << std::endl;
else
      std::cout << b << std::endl;
```

a good answer would be *"It prints out the smaller of the two values held by variables a and b."*

Do not give a line by line description of the code. The code should also not be evaluated on whether it compiles or not, your task is to determine the intent of the code, not the ability of the "programmer".

i. (2 marks) Consider the following code:

```
class node {
      private:
            int data;
            node* next;
      public:
            node(int data, node* next){
                  this->data = data;
                  this->next = next;
            }

            int get_data() const { return this->data; }

            bool p() const {
                  if (next == nullptr) return true;
                  if (data < next->get_data()) return false;
                  return next->p();
            }
};
```

Assume you have a linked list constructed of such `node`s. What property of the list does calling the function `p` on the head of the list test?

ii. (2 marks) In one sentence, explain in plain English what the following function does:

```
int w(const std::vector<int>& v) const {
      int n = 0;
      for (std::vector<int>::iterator itr = v.begin(); itr < v.end(); ++itr){
            n = n + *itr;
      }
      return n;
}
```

iii. (2 marks) In one sentence, explain in plain English what the following method does:

```
std::vector<int> x(std::vector<int> v){
      do {
            bool a = false;
            for (int i = 0; i < v.size() - 1; i++){
                  if (v[i] > v[i+1]){
                        int temp = v[i+1];
                        v[i+1] = v[i];
                        v[i] = temp;
                        a = true;
                  }
            }
      } while (!a);

      return v;
}
```

iv. (2 marks) In one sentence, explain in plain English what the following code does (assume you have a vector called v):

```
for (std::vector<int>::iterator itr = v.begin(); itr < v.end(); ++itr){
      *itr = *itr + 1;
}
```

v. (2 marks) In one sentence, explain in plain English what the following code does:

```
int z(std::vector<int> v){
      int n = 0;
      for (int i = 0; i < v.size(); i++){
            if (i % 2 == 0){
                  n += v[i];
            }
      }
}
```

Question 23 (10 marks)

This question concerns binary search trees.

i. (2 marks) In plain English, what is the traversal rule that Binary Search trees use to find the place to insert values in the tree and find values in the tree?
ii. (5 marks) Starting with an empty Binary Search Tree, insert the following values into the tree in the order presented: 10, 3, 9, 1, 15, 12, 11, 17. Draw the tree after each insertion.
iii. (1 marks) What is the average time to find an element in a Binary Search Tree with *n* nodes? Give your answer in asymptotic (big-oh) notation.
iv. (2 marks) Starting with an empty Binary Search Tree, what insertion order of the values: 10, 3, 9, 1, 15, 12, 11, 17 would produce worst-case behaviour of the Binary Search Tree.