1. (a) What is the value of `num` after this code has been run:

```
int  int1 = -5, int2 = 3;
float num = int1 / int2;
```

(b) What is the value of `*ptr` after this code has been run:

```
char *str1 = "Hello there Mary";
char *ptr = str1[6];
```

(c) Why won't the code below compile?

```
int num, *ptr = NULL;
ptr = new int;
&num = ptr;
```

(d) In what file is the class interface stored?

(e) Why do `<<` and `>>` have to be declared as 'friend' operators?

(f) What does it mean when we make member variables 'protected' instead of 'private'?

(g) What is a directed graph?

(h) If you require direct access, have unlimited RAM available, and have records with a key, what data structure would you use?

(i) What is the main difference between std::set and std::map?

(j) What is the main advantage of std::deque over std::vector for use with stacks and queues?

[1 mark each = 10]

2. (a) What does the term ‘dynamic binding’ mean?

   (b) What is a pure virtual method?

   (c) What is a queue?

   (d) What is a max heap?

   (e) What is an Adelson-Velski-Landis (AVL) tree?

$[2 + 2 + 2 + 2 + 2 = 10]$

3. (a) What are the relative advantages and disadvantages of an array compared to a linked list?

   (b) Write a pseudo-code algorithm for a straight selection sort for an array.

$[4 + 6 = 10]$

4. (a) What does LIFO stand for?

   (b) When you run a program, a stack is set up by the system for your program. What is it used for?

   (c) What are the only two methods—other than constructors and destructor—that are *always* defined for a Stack?

   (d) Write down the declaration for a stack *template*, which uses the standard template library *deque* for its data structure.

$[1 + 1 + 2 + 6 = 10]$

5. Draw UML diagrams showing

   (a) StandardAccount derived from BasicAccount.

   (b) NameClass containing m_first, m_second, m_surname, all of which are of class StringClass.

   (c) Node that contains data of type Data and a pointer to another node of type Node.

   [2 + 4 + 4 = 10]

6. (a) Name four methods or operators that must be coded in a Set class.

   (b) Write pseudo-code for a merge of two containers that performs an *intersection* of the containers.

   [4 + 6 = 10]

7. (a) Draw diagrams showing the steps involved in swapping the order of two nodes in a linked list.

   (b) Write down the C++ code for a method that does a linear search on a linked list of integers. You may assume all the normal Node methods exist. The method is to return true or false (function-wise) depending on whether the integer is found.

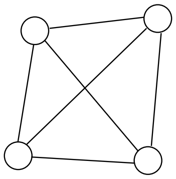
   [4 + 6 = 10]

8. (a) Draw a copy of the diagram below, with labels/additions showing the meaning of the terms:

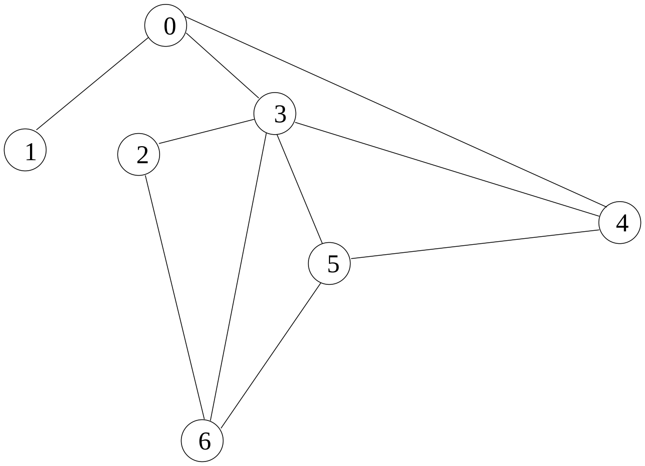
   i. edge

   ii. vertex

   iii. a tour



   (b) Draw the Adjacency List for the graph below:



[3 + 7 = 10]

9. (a) Write a paragraph briefly describing the observer design pattern.

   (b) Assuming that we have three classes derived from an Account class and that the user has chosen a number from the following menu:

```
TYPE OF ACCOUNT
0 - Standard
1 - Savers
2 - Overdraft
```

   Write the C++ code for the following factory method:

```
Account *GetAccount(int menuChoice);
```

[3 + 7 = 10]

10. (a) Given the following definition for a node:

```
class Node
{
public:
     Node () {m_left = m_right = NULL;}
    ~Node () {};
    Node *GetLeft   () {return m_left;}
    Node *GetRight  () {return m_right;}
    void  SetLeft   (Node *left) {m_left = left;}
    void  SetRight  (Node *right) {m_right = right;}
    bool operator < (Node *other) {return data < other-$>$data;}
    bool operator > (Node *other) {return data > other-$>$data;}
private:
    Node *m_left;
    Node *m_right;
    Data  m_data;
}
```

write C++ code for the following—recursive—Binary Search Tree method, that inserts a new node in the sub-tree below the parent node:

```
void TreeClass::Insert(Node *parent, Node *newNode);
```

You may assume that there will be no duplicate data.

(b) Draw the diagram of the 3-way, B-tree that results after the following numbers are inserted into an initially empty tree: 45, 23, 78, 14, 9, 22.

[5 + 5 = 10]

—— ◇◇◇ **END OF PAPER** ◇◇◇ ——