

## Chapter 10 Answers to some of questions

### True/False Questions

3) False                      4) True

### Multiple Choice questions

1) b)                          3) c)  
2) d)                          5) b)

### Short Answer Questions

5) the following code has a memory leak:

```
1   int *a, *b, c;  
2   a = new int;  
3   *a = 10;  
4   b = new int;  
5   *b = 12;  
6   a = b;  
7   delete a;  
8   delete b;
```

In line 6, the pointer **a** is re-referenced to the same memory location **b** is pointing at, losing information about the memory location with 10.

When **delete a** is performed, the memory with value 12 is released, the second **delete** is incorrect as it tries to release the memory location which was already released. Hence, memory location with 10 is never released – it is a memory leak.

6) The following code accesses memory it shouldn't access:

```
1   int *a, *b, c;  
2   a = new int;  
3   *a = 10;  
4   delete a;  
5   *a = 17;
```

In line 4, the memory location **a** was pointing at is released, so when we try to do the assignment in line 5, we are accessing memory we shouldn't access.