



University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

SEMESTER 2 EXAMINATION – 2011/2012: [SAMPLE SOLUTIONS](#)

COMP 20080

Computer Science for Engineers II

Prof. Alain Mille

Mr. J. Dunnion

Prof. L. Murphy*

Time allowed: 2 hours

Instructions for candidates

Answer ALL questions. Question 1 is worth 40 marks. All other questions are worth 3 marks each.

Write your answers in the Answer Books provided.

Instructions for invigilators

Loose Rough Work sheets are not to be distributed or used.

Use of calculators is prohibited.

SAMPLE SOLUTIONS

1. [40 marks: 20 marks for (a), 20 marks for (b)]

(a) **20 marks**

```
#include "Account.h"
Account::Account(int num, float bal)
{
    accountNumber = num;
    accountBalance = bal;
}

int Account::withdraw(float amount)
{
    if (amount > accountBalance) {
        return 0; // withdrawal cannot be completed
    }
    else {
        accountBalance = accountBalance - amount;
        return 1; // successful withdrawal
    }
}

void Account::deposit(float amount)
{
    accountBalance = accountBalance + amount;
}

int Account::getNumber()
{
    return accountNumber;
}

float Account::getBalance()
{
    return accountBalance;
}
```

(b) **20 marks**

```
#include "Account.h"
#include <iostream.h>
int main()
{
    int success; // to indicate whether withdrawal was ok
    int number = 1234; float balance = 1333.33;
    Account myAc(number, balance); myAc.deposit(666.67);
    cout << "account number " << myAc.getNumber() << " has a
current balance of " << myAc.getBalance() << ".\n";
    success = myAc.withdraw(987.65);
    if (success == 0) cout << "withdrawal failed\n";
    else cout << "withdrawal succeeded\n";
    cout << "account number " << myAc.getNumber() << " has a
current balance of " << myAc.getBalance() << ".\n";
    success = myAc.withdraw(5000);
    if (success == 0) cout << "withdrawal failed\n";
    else cout << "withdrawal succeeded\n";
    cout << "account number " << myAc.getNumber() << " has a
current balance of " << myAc.getBalance() << ".\n";
}
```

SAMPLE SOLUTIONS

2. (a) and (c)
3. False
4. (b) and (c)
5. False
6. (b)
7. (a) and (c)
8. (c) and (d)
9. False
10. (a), (b) and (c)
11. (b) and (d)
12. True
13. (a) and (b)
14. True
15. In compiler implementations, a typical loop optimisation technique is to replace a loop which has a fixed number of iterations with the “unrolled” sequence of loop body statements.
16. (b) and (c)
17. (b)
18. False
19. (a), (c) and (d)
20. (c)
21. True