SIT232 Object-Oriented Development Exam Solution – Trimester 1, 2014

This solution covers Section B (short answer) of the examination paper only. Section A (multiple choice) is drawn from a large pool of questions which are marked automatically by Scanning within DSA. To maintain the value of the pool of multiple choice questions, no reproduction of those questions or answers is made available to students.

Note that all answers provided are only for background, it is not expected that a student will have the exact same answer, or necessarily the same level of detail.

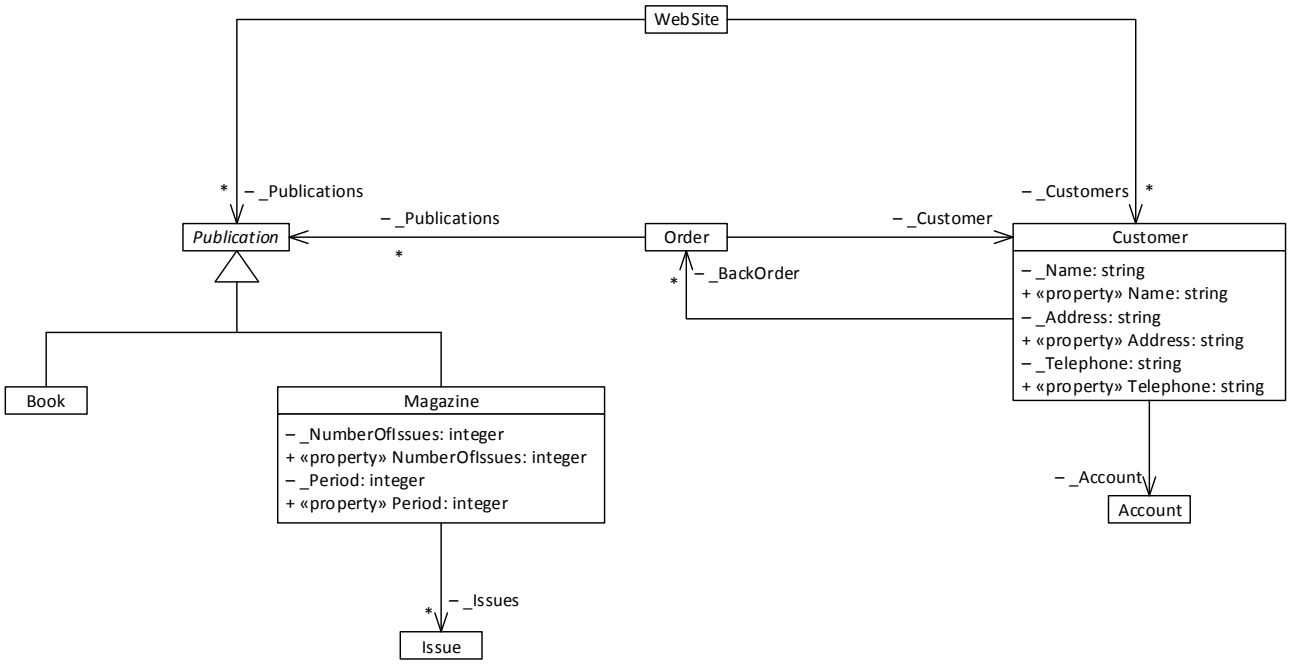
Question a.

a) Consider the following problem statement:

You have been hired by a major online retailer to develop a new software application to receive and manage home delivery of publications (books and magazines). The customers of the retailer can register their details using the web site including name, address, and telephone numbers. They can then browse the publications available for sale, including individual titles (books) and subscription items (number of issues over a defined period), adding those items to their shopping trolley. Once the order is complete, the retailer staff will package any individual books and current edition of a magazine and deliver them to the customer. The total cost of goods purchased, plus a supplemental charge of $5 per book or $10 per subscription, is charged to the customer's account. A record of any items that are not available are kept in the database as backorders and are automatically added to the next order from the customer.

Your task is to prepare a UML class diagram illustrating the concepts in the problem statement above. Your class diagram must demonstrate the following elements:

- Attributes, properties, and methods;
- Collections;
- Abstract classes;
- Inheritance;
- Associations (any type); and
- Cardinality.



Note that the above UML is only one example solution, many others will be acceptable.
Marking scheme (15 marks):

7.0 marks (1.0 mark each) – the following elements are applied appropriately in the UML: attributes, properties, methods, collections/cardinality, abstract classes, inheritance, and associations.

6.0 marks (2.0 marks each) – the following elements are represented appropriately in the UML: publications (including types of publications), customers, and orders (including current and backorders).

2.0 marks – the notation in the UML is otherwise correct

Question b.

b) For this task you are required to write C# code satisfying the following requirements:

- An abstract class named Building which has:
  - Two string attributes for the address and phone number;
  - Read-only properties encapsulating the above attributes;
  - A method ChangePhone which allows only derived classes to provide a new value for the phone number attribute;
  - A custom constructor which takes address and phone number parameters and initialises the attributes defined above; and
  - A ToString() method which returns a string containing the above data in the format:

    address: *phone number*
- An class named RentedBuilding which inherits from the Building class and has:
  - A decimal attribute for the rental per week;
  - A read-only property encapsulating the above attribute;
  - A custom constructor which takes address, phone number, and rental per week, passes the address and phone number to the base class constructor and initialises the rental per week attribute defined above;
  - A ToString() method which returns a string containing all of the stored data (with rental in currency format) in the format:

    address: *phone number* (*rental per week*)

```
abstract class Building
{
    private string _Address;
    public string Address
    {
        get { return _Address; }
    }

    private string _PhoneNumber;
    public string PhoneNumber
    {
        get { return _PhoneNumber; }
    }

    protected void ChangePhone(string value)
    {
        _PhoneNumber = value;
    }

    public Building(string address, string phoneNumber)
    {
        _Address = address;
        _PhoneNumber = phoneNumber;
    }

    public override string ToString()
    {
        return string.Format("{0}: {1}", _Address, _PhoneNumber);
    }
}
```

```
class RentedBuilding : Building
{
    private decimal _RentalPerWeek;
    public decimal RentalPerWeek
    {
        get { return _RentalPerWeek; }
    }

    public RentedBuilding(string address, string phoneNumber, decimal rentalPerWeek)
        : base(address, phoneNumber)
    {
        _RentalPerWeek = rentalPerWeek;
    }

    public override string ToString()
    {
        return string.Format("{0} ({1:c}", base.ToString(), _RentalPerWeek);
    }
}
```

Marking scheme (15 marks):
Building class:

- *1.0 mark – class is declared with abstract keyword*
- *2.0 marks – address and phone number attributes and read-only properties correct*
- *2.0 marks – constructor accepts parameters and initialises attributes correctly*
- *2.0 marks – ChangePhone method has protected accessibility and is otherwise correct*
- *2.0 marks – ToString returns correctly formatted string and is otherwise correct*

RentedBuilding class:

- *1.0 mark – class inherits from Building class*
- *1.0 mark – rental per week attribute and read-only property correct*
- *2.0 marks – constructor accepts and initialises rental and passes remaining parameters to base*
- *2.0 marks – ToString returns correctly formatted string, uses base.ToString, and is otherwise correct*

Penalise up to 2.0 marks is there are clear problems with the syntax.
