

# ***SAMPLE PAPER***

**Institute of Systems Science  
National University of Singapore**

## **GRADUATE DIPLOMA IN SYSTEMS ANALYSIS**

### **SA Term 1 Examination Sample Paper 2: Enterprise Solutions Design and Development**

**Matriculation Number:**

*(fill in your matriculation number here)*

### **Instructions for Paper**

Date: Day, Date  
Time: 9.00am  
Duration: 3 hours (9.30 a.m. to 12.30 p.m.) excluding reading time  
Place: NUS MPSH4

This is an **open book** examination.

You are given **30 minutes** to read the questions before the examination starts. You are **NOT** allowed to use any writing instrument during the reading time.

1. Read all instructions before answering any of the examination questions.
2. Write your matriculation number on the top of this front page in the box provided.
3. Complete the front cover of each answer book you use. The blank lines on the front cover to be filled in as follows:

Matriculation / Registration No:	= (Your matriculation number)
Module Code/Title:	= Paper 2: Enterprise Solutions Design and Development
Semester:	= SA Term 1
Number of books handed in:	= (indicate the number of booklets you have used for each section.)

4. Write your matriculation number on **ALL** answer books you use.
5. This examination paper consists of **three (3)** sections and **xx (x)** questions. You are to answer **ALL** questions.
6. The total marks for this examination paper is **xx (xx)** marks. **xx (xx)** marks will be taken from the continuous assessment.
7. An answer to a question must begin on a new page.
8. **Use a SEPARATE Answer Book for each SECTION.**
9. After completing the paper, tie your answer books together according to **sections**. Make sure that the questions for each section are identified on the front page of each answer book.
10. The question paper is to be submitted together with the answer books. **You are not allowed to take the question paper with you.**
11. Scrap paper may be used as required, but only answers written in the answer book will be considered for credits.
12. Use a pen for writing your answers. Pencil may be used only for drawing diagrams.
13. Calculators may be used if required.
14. State clearly any assumptions you make in answering any question where you feel the requirement is not sufficiently clear.

## Section A [20 marks]

### Fundamental of Programming using C# .NET

All questions in this section are based on the ABC Stores Case Study given in **Appendix A**. Also refer to **Appendix B** for some APIs that help you answer the programming question.

#### Question 1 [20 Marks]

Write a C# **console program** that would be used by the counter salesperson to print out the invoice slip.

Your program should read the following information from the console:

- Item Code (string)
- Item Price (decimal)
- Item Quantity (integer)
- Is the shopper a loyalty member? (char – Y/N)
- Membership Number [to be input if response to (d) is Y]

*TIPS: The program should:*

- Prompt for Item data (Item Code, Item Price, Item Quantity).
- After reading the data, if sales person enters 'Y', accept data for the next item.
- Repeat above process for all the items.
- For programming purpose if you intend using arrays, you can make appropriate assumption on the maximum array size.
- Compute the cost for each item and print each item's detail (eg. midweek discount and total price).
- Also compute and print the gross total, membership discount, GST and payment amount at the bottom of the invoice slip (as per the format shown in annexure).

You may assume:-

- The counter salesperson would enter all the data **correctly** and hence your program **need not** perform input data validations.
- You **are permitted** to use the ISS.RV.LIB libraries for reading the data input from console.

Your answer should include:-

- You are expected to write a static class with a main method that would perform the above task. You **need not** demonstrate OOP approach for this question.
- A syntactically correct program that meets the specifications is essential.
- In addition credit would be given if you are able to accomplish the followings:
  - Organize your program modularly with appropriate static methods.
  - Demonstrate ability to format the invoice slip well.
  - Demonstrate best practices such as using good naming standards and clear presentation of program.

~~~~~ *End of Section A* ~~~~~

## Section B [25 marks]

### Object – Oriented Programming using C# .NET

#### QUESTION 2. (10 marks)

(a)

```
using System;

public class Person {
    private string name;
    public string Name {
        get { return (name); }
    }
    public Person(string n) {
        name = n;
    }
    public virtual void Show() {
        Console.WriteLine("Person is {0}", Name);
    }
}

public class Employee : Person {
    private string supervisor;
    public string Supervisor {
        get { return (supervisor); }
    }
    public Employee(string n) : base(n) {
        supervisor = "Big Boss";
    }
    public new void Show() {
        Console.WriteLine("Employee is {0}", Name);
    }
}

public class App {
    public static void Main() {
        Person[] employees = {
            new Person("Ah Kow"),
            new Person("Ah Ter"),
            new Employee("Ah Yeow"),
            new Employee("Ah Beng")
        };
        foreach (Person p in employees) {
            p.Show();
        }
    }
}
```

What is the outcome/output of compiling and/or executing the program above? Explain the reason for this outcome.

(b) (2 marks)

```
using System;

public class Person {
    private string name;
    public string Name {
        get { return (name); }
    }
    public Person(string n) {
        name = n;
    }
    public virtual void Show() {
        Console.WriteLine("Person is {0}", Name);
    }
}

public class Employee : Person {
    private string supervisor;
    public string Supervisor {
        get { return (supervisor); }
    }
    public Employee(string n) : base(n) {
        supervisor = "Big Boss";
    }
    public override void Show() {
        Console.WriteLine("Employee is {0}", Name);
    }
}

public class App {
    public static void Main() {
        Person[] employees = {
            new Person("Ah Kow"),
            new Person("Ah Ter"),
            new Employee("Ah Yeow"),
            new Employee("Ah Beng")
        };
        foreach (Person p in employees) {
            p.Show();
        }
    }
}
```

What is the outcome/output of compiling and/or executing the program above? Explain the reason for this outcome.

(c) (2 marks)

```
using System;

public struct Point {
    public int x;
    public int y;
    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
    public override string ToString() {
        return(String.Format("[Point:{0},{1}]", x, y));
    }
}

public class Point2 {
    public int x;
    public int y;
    public Point2(int x, int y) {
        this.x = x;
        this.y = y;
    }
    public override string ToString() {
        return(String.Format("[Point2:{0},{1}]", x, y));
    }
}

public class App {
    public static void Main() {
        Point a = new Point(3,4);
        Point b = a;
        a = new Point(5,6);
        Console.WriteLine(a);
        Console.WriteLine(b);
        Point2 p = new Point2(23,24);
        Point2 q = p;
        p = new Point2(25,26);
        Console.WriteLine(p);
        Console.WriteLine(q);
    }
}
```

What is the outcome/output of compiling and/or executing the program above? Explain the reason for this outcome.

(d) (2 marks)

```
using System;

public struct Point3 {
    public int x;
    public int y;
    public Point3() {
        this.x = 1;
        this.y = 1;
    }
    public override string ToString() {
        return(String.Format("[Point3:{0},{1}]", x, y));
    }
}

public class App {
    public static void Main() {
        Point3 p = new Point3();
        Console.WriteLine(p);
    }
}
```

What is the outcome/output of compiling and/or executing the program above? Explain the reason for this outcome.

(e) (2 marks)

```
using System;

public class Person {
    private string name;
    public string Name {
        get {
            return (name);
        }
    }
    public Person(string n) {
        name = n;
    }
    public virtual void Show() {
        Console.WriteLine("Person is {0}", Name);
    }
}

public class App {
    public static void Main() {
        Person p = new Person("Ah Ter");
        p.Name = "Mrs" + p.name;
        Console.WriteLine(p);
    }
}
```

What is the outcome/output of compiling and/or executing the program above? Explain the reason for this outcome.

**QUESTION 3. (11 marks)**

- (a) Write a **StockItem** class with **productCode (string)**, **description (string)**, **quantity (int)** and **unitPrice (double)** attributes.
- (b) You must use the appropriate C# mechanisms to ensure representational independence; and you may assume that the **quantity** and **unitPrice** attributes will be updated externally, but not the **productCode** and **description** attributes.
- (c) Provide an appropriate constructor.
- (d) Provide a method **GetValue ()** or property **Value** to return the current stock value of a **StockItem** instance (based on its **quantity** and **unitPrice**).

**QUESTION 4. (4 marks)**

Consider the following additional methods **Buy ()** and **Sell ()** for the **StockItem** class.

```
class StockItem {  
    ...  
  
    public void Buy(int n) {  
        quantity = quantity + n;  
    }  
  
    public void Sell(int n) {  
        if (quantity < n)  
            Console.Error.WriteLine("Cannot sell more than what we have");  
        else  
            quantity = quantity - n;  
    }  
  
    ...  
}
```

- (a) Give one reason why writing an error message for an invalid quantity in the **Sell ()** method might not always be ideal.
- (b) Enhance the **Sell ()** method so that it is more flexible in coping with errors.
- (c) Show how you would make a call to the enhanced **Sell ()** method.

~~~~~ **End of Section B** ~~~~~



## Section C [25 marks]

### SQL Programming and User Interface Programming using C# .NET

#### Course Registration System Case Study

You may assume that you are working on SQL server. State clearly any other assumptions made in answering Question 5 and 6.

**Table Name: Course**

| Field Name         | Type          | Description  |
|--------------------|---------------|--|
| CourseCode         | Varchar(10)   | course code is unique (example JAVA, PM, etc)  |
| CourseDescription  | Varchar (30)  | a short and easily to understand course name (example Java Programming, Project Management, etc)   |
| CourseDuration     | Integer       | the number of days of a run of the course  |
| CourseFee          | Numeric (6,2) | the actual course fee per student in dollars and cents   |
| CourseMaxClassSize | Integer       | the maximum number of students that can be accepted into a run of this course <ul style="list-style-type: none"> <li>every run of the same course has the same maximum class size</li> </ul> |

**Table Name: CourseRun**

| Field Name           | Type        | Description   |
|----------------------|-------------|---|
| CourseCode           | Varchar(10) | course code of this course run  |
| CourseStartDate      | DateTime    | the date of the first day of a run of this course <ul style="list-style-type: none"> <li>a course will not have two runs on the same start date</li> <li>a course may have one or more runs in a year or none at all</li> </ul> |
| TotalStudentEnrolled | Integer     | The number of students enrolled for this run at the current point in time. <b>Add 1 when a student enrolls and minus 1 when a student withdraws.</b>  |

#### Question 5 [15 marks]

- List all possible constraints you would want to impose on the tables above. You are **NOT REQUIRED** to write SQL statement. (4 marks)
- Write a store procedure to display a list of course code and course description with course fees less than \$1500. (3 marks)
- Write an SQL statement to display all course code and average number of students enrolled (per each run of the course). (3 marks)
- Write an SQL statement to display a list of course code and course description that have more students enrolled than the course code 'JAVA'. (5 marks)

**Question 6 [10 Marks]**

- a) Design a windows forms screen called **ClassForm** that would provide an interface for the user to create essential data for CourseRun record. The Designed form should have the following features:

- The form should have a neat look & feel using the most appropriate visual controls.
- The form should display only one record at a time.

State clearly the type of control you used in the form to capture the various data elements.

**(6 marks)**

- b) Describe with example how the following features can be used to in your screen design. You are not required to write any code.

- Combo Box
- List Box

**(4 marks)**

~~~~~ *End of Section C* ~~~~~

~~~~~ *End of Sample Paper 2* ~~~~~