Rewarding Learning

**ADVANCED SUBSIDIARY (AS)**
**General Certificate of Education**
**2017**

**Centre Number**

**Candidate Number**

# Software Systems Development

## Unit AS1:

## Introduction to Object Oriented Development

A1S11

## [A1S11]
## MONDAY 15 MAY, AFTERNOON

**TIME**

2 hours.

**INSTRUCTIONS TO CANDIDATES**

Write your Centre Number and Candidate Number in the spaces provided at the top of this page.
Write your answers in the spaces provided in this question paper.
Answer **all six** questions.

**INFORMATION FOR CANDIDATES**

The total mark for this paper is 100.
Figures in brackets printed down the right-hand side of pages indicate the marks awarded to each question or part question.

**ADVICE TO CANDIDATES**

You are advised to take account of the marks for each part question in allocating the available examination time.

| For Examiner's use only | | | |
|---|---|---|---|
| **Question** | **Marks available** | **Marks** | **Remark** |
| 1 | 9 | | |
| 2 | 23 | | |
| 3 | 23 | | |
| 4 | 14 | | |
| 5 | 24 | | |
| 6 | 7 | | |
| **Total** | **100** | | |

| Examiner Only | |
|---|---|
| Marks | Remark |

**1** Complete the following statements about an object-oriented programming environment by inserting the appropriate words or phrases from the list given below.

overloading interfaces inheritance classes polymorphism

signatures overriding implements methods late binding

new Object early binding object instantiation

**(i)** _____ is the creation of an object to the design of a class.

**(ii)** _____ is the base building block of an object-oriented system and all _____ are derived from it.

**(iii)** _____ occurs where two or more methods have the same name but different _____, even if their return types differ.

**(iv)** Customising a super / base method within a derived/sub class is known as _____.

**(v)** Multiple inheritance is implemented in C#/Java through the use of _____.

**(vi)** A primary concept of object-oriented programming is _____. It allows sub/derived class methods to be invoked through a super / base class reference during run-time. This is enabled through _____. [9]

**2** An artist runs a restoration service where donated items or those retrieved from recycling, are restored or turned into alternative reusable items. These are then offered for sale through upmarket retailers. Restorations fall into three main categories, A – Lamps, B – Tables and C – Sundry.

The production cost is the total of the labour cost, materials cost and specialism cost. The labour cost is the number of hours multiplied by the rate of £20 per hour. Specialism costs occur for outsourced work when required.

The sale price is determined by increasing the production cost by 40%.

**(a)** Complete the design of a class called **Job** shown below.

Ensure the creation of:

- A constant variable to hold the value 20 for the hourly rate;
- **Get** and **Set** (Properties / Methods) for the field **noHours** only;
- **Methods** to determine appropriate costs.

```
class Job{
    private   int        jobNo;
    private   String     description;
    private   char       category;
    private   DateTime   jobDate;
    private   DateTime   saleDate;
    private   int        noHours;
    private   double     materialCost;     // examples – paint, material
    private   double     specialismCost;

//  constant variable to hold the value 20 for the hourlyRate.
```

_____

_____ [2]

| Examiner Only | |
|---|---|
| Marks | Remark |
| | |

```
public Job(){
      private    jobNo = 0;
      private    description = null;
      private    category = ' ';
      private    jobDate = new DateTime();      // sets today's date
      private    saleDate = default(DateTime);  // sets date to 01/01/0001
      private    noHours = 0;
      private    materialCost = 0.0;
      private    specialismCost = 0.0;
}
//    GET and SET          (Property/Methods) for the field noHours only
```

[4]

// A method to determine the **productionCost** due on a job.

[3]

| Examiner Only | |
|---|---|
| Marks | Remark |
| | |

// A method to determine the **salePrice** for a job.

_____

_____

_____

_____

_____

_____ [4]

**(b)** A program using the class Job has the following line of code:

Job [ ] jobArray = **new Job[150];**

**(i)** Explain the terms in bold.

_____

_____

_____

_____

_____

_____ [2]

| Examiner Only | |
|---|---|
| Marks | Remark |

**(ii)** The array has been populated and currently has the details of **72** finished items recorded. Write the section of code which will display details of the **lamps available** for sale. Finally output the **overall total** for the production costs for these **unsold** items.

The category for lamps is **A**.

Available items have the default saleDate of 01/01/0001.

Assume a toString method exists in the class Job that returns jobNo, description, production cost and the sale price.

[8]

| Examiner Only | |
|---|---|
| Marks | Remark |
| | |

**BLANK PAGE**

**(Questions continue overleaf)**

**3** Sorting is a common activity conducted on data in information processing.

**(a) (i)** Name and describe a simple sorting method.

[5]

| Examiner Only | |
|---|---|
| Marks | Remark |
| | |

**(ii)** Using the following set of numbers illustrate the contents after each pass of your chosen sort.

Indicate the number of swaps at each pass.

5, 16, 11, 7, 26

[5]

| Examiner Only | |
|---|---|
| Marks | Remark |

**(b)** C# and Java provide a SORT class to facilitate sorting requirements.

Examiner Only

| Marks | Remark |
|---|---|
| | |

Complete the sentence below.

**(i)** To allow use of the SORT facility a class header must implement

the _____ called _____.

[2]

**(ii)** For the following class Product, a list of product details is required where the product category is in alphabetical order and the price within each category is in ascending order.

**C#**

```
class Product : IComparable{
        private int         productNo;
        private String      description;
        private String      category;
        private int         noOfProduct;
        private double      price;
```

**java**

```
class Product implements Comparable{
        private int         productNo;
        private String      description;
        private String      category;
        private int         noOfProduct;
        private double      price;
```

Write the method that will accomplish this requirement by comparing two product objects and return an integer that is:

negative if the first object is 'less than' the second object;
zero if the objects are equal;
positive if the first object is 'greater than' the second object.

*Note that the class **String** facilitates a SORT with a **compareTo** method that returns an integer to indicate the alphabetical order of the two String objects being compared.*

*Hint: the price only needs to be considered if the two product items have the same category.*

| Examiner Only | |
|---|---|
| Marks | Remark |

[11]

**BLANK PAGE**

**4** Details of stock for sale are defined in a class Stock as shown below:

```
class Stock
{
    private    String    stockNo;      // maximum 8 chars
    private    String    model;        // maximum 12 chars
    private    int       qtyInStock;   // range 0 – 6
    private    double    price;        // range 1200 – 35000
}
```

**(a)** Write the output statement which will **display** the details of an item of stock including the stock value. Ensure the information is formatted appropriately.

_____

_____

_____

_____

_____

_____ [3]

**(b)** The stock will be stored in a binary file for subsequent processing. What amendment must be made to the class definition to facilitate reading and writing of stock objects to and from the file?

_____

_____ [1]

Examiner Only

| Marks | Remark |
|---|---|
| | |

**(c)** Below is a section of code to deal with a file of stock objects. Choosing either the C# or Java example explain what you understand by the emboldened terms.

| Examiner Only | |
|---|---|
| Marks | Remark |
| | |

*Section of C# code*

```
int size=0;
Stream strm;
try
{
strm = File.OpenRead("Stock.dat");
BinaryFormatter bf = new BinaryFormatter( );

try
{
      while (strm.Position < strm.Length)
      {
            arrayStock[size] = (Stock)bf.Deserialize(strm);
            size++;
      }
      strm.Close();
}
catch
```

*Section of Java code*

```
int size = 0;
try{
   FileInputStream strm = new FileInputStream("Stock.dat");
   ObjectInputStream ois = new ObjectInputStream(strm);

   try
   {
         while (strm.Position < strm.Length)
         {
               arrayStock[size] = (Stock)ois.readObject();
               size++;
         }
         strm.Close();
   }
   catch(
```

| Examiner Only | |
|---|---|
| Marks | Remark |

[8]

**(d)** Exception Handling must be implemented for the Input/Output of data to files.

Name the two Exceptions which would be applicable for the section of code given in part **(c)**.

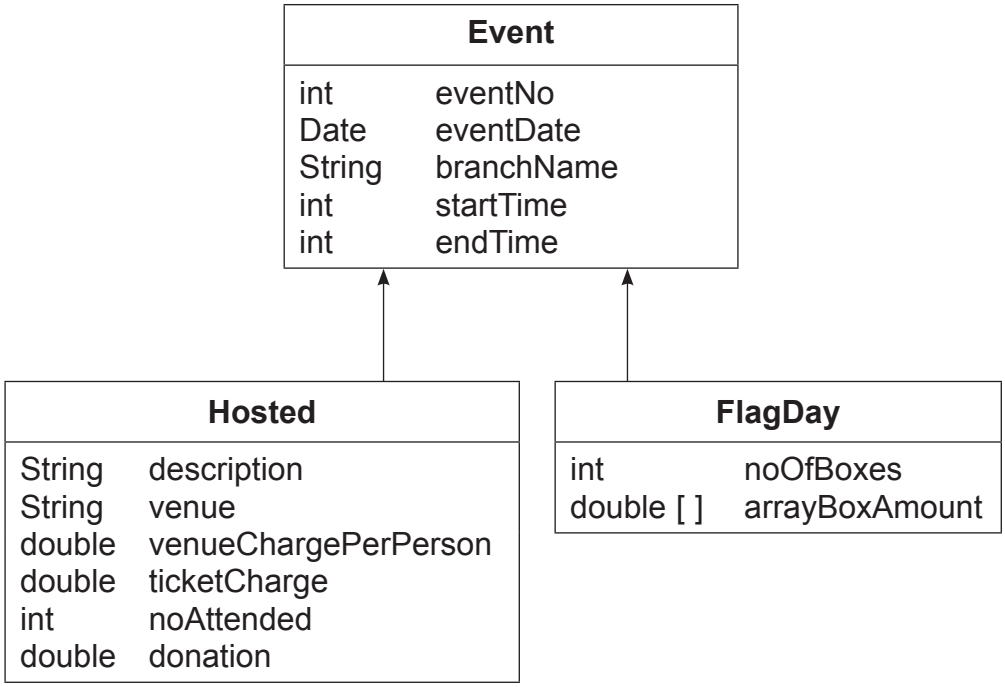
[2]

| Examiner Only | |
|---|---|
| Marks | Remark |

**5** A charity's fund-raising section supports a variety of events run by volunteer branches throughout the country. Common events are flag days, breakfasts, lunches, dinner dances and sporting events.

The charity holds information about each event for the purposes of advertising and the tracking of income. The volunteer branch deals with the planning details.

Inheritance diagram for the charity fund-raising section.




The header for the class **Event** has been defined as follows:

**abstract** class Event

**(a)** Explain the term in bold indicating why it has been applied to the class Event.

_____

_____

_____

_____ [2]

Assuming the class Event has been designed with the following elements

- field definitions;
- default and field/parameterised constructors;
- GET and SET (Properties/Methods);
- toString() method.

**(b)** Write the code for a method **HostedIncome** that will return the income generated from a **Hosted** Event.

Income is calculated as profit on a ticket, multiplied by noAttended, plus donations. The profit on a ticket is calculated as the ticketCharge minus the venueChargePerPerson.

_____

_____

_____

_____

_____

_____ [4]

| Examiner Only | |
|---|---|
| Marks | Remark |
| | |

**(c)** Write the code for the derived class **FlagDay** giving the following:

- class header;
- field definitions;
- field constructor (parameterised constructor);
- GET and SET (Property/Method) for noOfBoxes only.

Note that the noOfBoxes value is used to instantiate the arrayBoxAmount.

[11]

| Examiner Only | |
|---|---|
| Marks | Remark |
| | |

**(d)** Write the code for the method **FlagDayIncome( )** in the class FlagDay that will total up and return the income from the boxes for a Flag Day.

[7]

| Examiner Only | |
|---|---|
| Marks | Remark |
| | |

**6** Information from a system takes many forms such as reports (operational, analytical, financial) and results of enquiries.

The charity described in **question 5** has a program that tracks up to a maximum of 300 events across the country in an array called **arrayEvent** and currently holds data for 84 events denoted by **currentSize**.

A common enquiry from a fund-raising branch is for the total amount raised from their events to date.

Complete the body of code for the method defined below that will return the total income for an individual fund-raising branch. Call the method HostedIncome or FlagDayIncome as appropriate.

```
public double BranchIncome (Event [ ] arrayEvent, String
requiredBranch, int currentSize)
{
```

[7]

| Examiner Only | |
|---|---|
| Marks | Remark |
| | |

**THIS IS THE END OF THE QUESTION PAPER**

225172