INSTITUTE OF TECHNOLOGY
TALLAGHT
Higher Certificate in Science
Bachelor of Science
Bachelor of Science (Honours)

Computing

Full Time

Semester Three : January 2014

Software Development 3

Internal Examiners
Ms. Patricia Magee

External Examiners
Mr. Dermot Logue

Day **Wednesday**
Date **15th January 2014**
Time **15.30-17.30**

Instructions to Candidates

Answer Question One and any two other questions

Question 1 **(40 Marks)**

The following code partially defines an abstract superclass called Product, which is contained in a package called products. The Product class is part of a Product Management System, which calculates prices of various types of products. The Product class has two subclasses Electronics and Book which are also partially defined in the code below:

```
package products;

public abstract class Product {

    private double regularPrice;

    public Product(double regularPrice) {
        this.regularPrice = regularPrice;
    }
    public double getRegularPrice() {
        return regularPrice;
    }

    public void setRegularPrice(double regularPrice) {
        this.regularPrice = regularPrice;
    }
    public abstract double computeSalePrice();
}

package products;

public class Electronics extends Product{

    private String manufacturer;

    public Electronics(double regularPrice,String manufacturer) {
        super(regularPrice);
        this.manufacturer = manufacturer;
    }
}
```

```
package products;

public class Book extends Product{
    private String publisher;
    private int yearPublished;

    public Book(double regularPrice,String publisher,int yearPublished)
    {
        super(regularPrice);
        this.publisher = publisher;
        this.yearPublished = yearPublished;
    }
}
```

Write the java code for the following:

a) A class called **TV** which is a type of electronics product and exists in the same package. This class has one additional member variable size which represents the screen size of a TV. You need to provide the code for member variable declaration and for a constructor.

(6 Marks)

b) A **computeSalePrice()** for each class which overrides the method in the Product class. This method calculates and returns the sale price for each product. Electronic goods are sold at 60% of the regular price. Books are sold at 50% of the regular price and TVs are sold at 80% of the regular price.

(6 Marks)

c) Write the code for a test class, TestProduct to do the following:

i) This class should declare and initialise an array of type Product to hold the following 4 instances of the Book and TV classes:

Product	**Price**	**Manufacturer**	**Size**
TV	1000	Samsung	30
TV	1500	Sony	43
Product	**Price**	**Publisher**	**Year Published**
Book	34	Sun Press	2009
Book	15	Wiley	2013

(4 Marks)

ii) Display the regular and sale prices for all products using polymorphic method calls.

(8 Marks)

iii) Calculate and display the total regular and sale prices for all products. Sample output is shown below:

```
Item number 1, Regular price = €1000.0, Sale price = €800.0
Item number 2, Regular price = €1500.0, Sale price = €1200.0
Item number 3, Regular price = €34.0, Sale price = €17.0
Item number 4, Regular price = €15.0, Sale price = €7.5
Total Regular Price = €2549.0
Total Sale Price = €2024.5
```

(4 Marks)

d) In the context of the example above explain the purpose of abstract classes and methods in Java.

(6 Marks)

e) The classes above are all located in the same package called products. Explain the concept of packages in Java. How are they used?

(6 Marks)

Question 2 **(30 Marks)**

The following code defines a class called Employee which represents employees within a Company. A Company contains a number of different employees.

```
public class Employee {

	private String name;
	private String idNumber;

	public Employee(String name, String id) {
		this.name = name;
		this.idNumber = id;
	}

	public void setName(String name) {
		this.name = name;
	}

	public String getName() {
		return name;
	}

	public void setidNumber(String id) {
		idNumber = id;
	}

	public String getidNumber() {
		return idNumber;
	}
}
```

Write the Java code for the following:

a) A class called Company with the following implementation details:
   - Company has two data members: name (String) and the list (ArrayList) of employees within the Company.
   - A constructor that initialises both data members using a parameter list.
   - A showList() method that displays relevant information for the company and for each employee.

(16 Marks)

b) A test class to perform the following tasks:
   - Create a Company object reference for a company called Excul that has the following employees:

Employee Name	**Employee ID**
John	R1001
Paul	R1002
Ben	R1003
Kevin	R1004

   - Call the appropriate method to display the company name and the relevant information for each employee.

(9 Marks)

c) Explain what you understand by the term **"Composition"** in relation to the **Company** and **Employee** classes.

(5 Marks)

Question 3 **(30 Marks)**

The following code consists of 2 classes: a **Product** class and a **TestProduct** class for a Retail system.

```
public class Product
{
    private double price;

    public Product(double price)
    {
        this.price =price;
    }
    public void increasePrice(double amt)
    {
      price += amt;
    }

    public double getPrice()
    {
        return price;
    }
}
public class TestProduct
{
    public static void main(String[] arg)
    {
        Product p1 = new Product(350);
        p1.increasePrice(-25);
    }
}
```

a) The method increasePrice can only accept a positive value as a parameter. Write an exception class **PriceIncreaseException** that prints an error message "The price increase amount must be positive".

(5 Marks)

b) Modify the code in the Product class such that: If a negative amount is passed into the increasePrice() method, a **PriceIncreaseException** exception is ***thrown*** within the increasePrice method and is ***caught*** in the calling method (main).

(7 Marks)

c) Modify the above code (again the original code as specified above) such that: If a negative amount is entered, the **PriceIncreaseException** exception is handled *locally* within the increasePrice method.

(6 Marks)

d) Give an overview of the different kinds of Exceptions that may be encountered in Java, explaining which ones must be handled and which ones are not.

(8 Marks)

e) Explain the use of the Finally keyword in relation to Exceptions.

(4 Marks)

Question 4 **(30 Marks)**

a) Using the example of an integer in Java (int), describe the effect of three access modifiers on that integer, with reference to packages.

(9 Marks)

b) The keyword 'final' can be placed before variables, methods and classes. Illustrating your answer with examples, show how this affects the variable, method and class.

(9 Marks)

c) Write a piece of code to declare class called Outer that has a nested class called Inner. Both classes have constructors that accept an integer as a parameter, using it to initialise member variables. Inner has a method called 'add', which adds these two integers, and displays the result.

(12 Marks)