CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Autumn Examinations 2010/11

Module Title: Object-Oriented Programming 2

Module Code: **COMP 7013**

School: Computing

Programme Title:
Bachelor of Science in Computing – Year 3
Bachelor of Science (Honours) in Software Development & Computer Networking – Year 2
Bachelor of Science (Honours) in Software Development – Year 2

Programme Code: **KCOMP_7_Y3**
KSDEV_8_Y2
KDNET_8_Y2

External Examiner(s): **Mr. Peter Given**
Internal Examiner(s): **Ms. D. M. Dunlea, Mr. Denis Long**

Instructions: **Section A**: Question 1 must be answered.
Section B: Choose 2 questions from this section.

Duration: 2 Hours

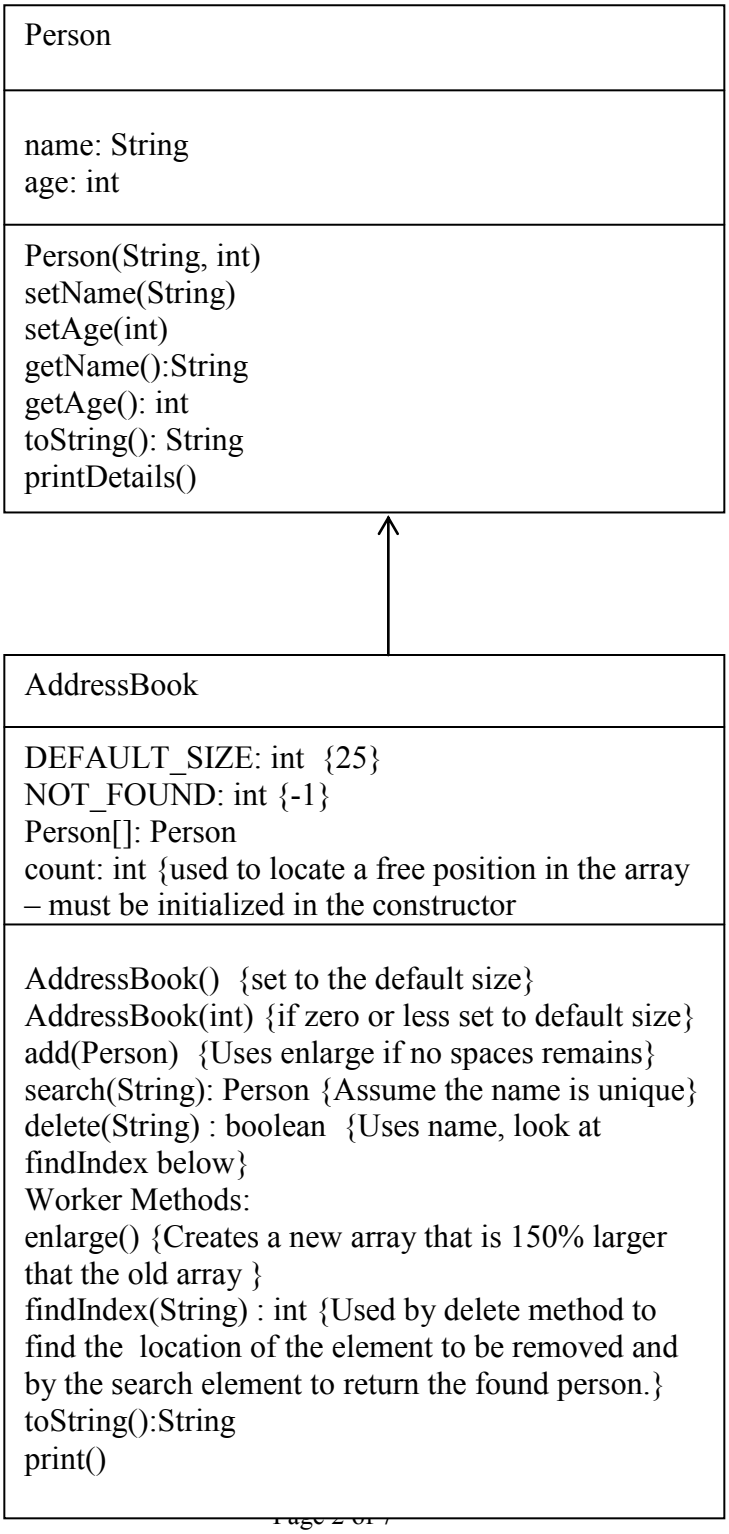
Sitting: Autumn 2011

Requirements for this examination:

Note to Candidates: Please check the Programme Title and the Module Title to ensure that you have received the correct examination paper.
If in doubt please contact an Invigilator.

Section A

Q1. (a) Write the code for the following classes. [30 Marks]



(b) Create an application to enter 5 people you know into the address book. Once the details have been entered search the address book and find and print the details for the following:

the average age for the people entered;

the youngest person and

the oldest person in the address book. [10 marks]

Section B

Q2. (a) Please answer True or False to the following questions:

1. To use the Math class, the first step is to make an instance of it.
2. You can mark a constructor with the **static** keyword.
3. Static methods don't have access to instance variable state of the 'this' object.
4. It is good practice to call a static method using a reference variable.
5. Static variables could be used to count the instances of a class.
6. Constructors are called before static variables are initialized.
7. MAXSIZE would be a good name for a static final variable.
8. A static initializer block runs before a class's constructor runs.
9. If a class is marked final, all of its methods must be marked final.
10. A final method can only be overridden if its class is extended.
11. There is no wrapper class for boolean primitives.
12. A wrapper is used when you want to treat a primitive like an object.
13. The parseXxx methods always return a String.
14. Formatting classes (which are decoupled from 1/0), are in the java.format.package.

[7 Marks]

(b) Write a class for IDCard that inherits from the card class below. An IDCard is a card that holds a name and an ID number. The ID number of an IDCard is guaranteed to be unique. It should be possible to get the name and identifier of a card. To achieve this you should implement a method on IDCard that returns a string showing the name and id of the card as follows:

```
Cardholder:  Susan
 ID: 12
```

```
public class Card
{
        private String name;

        public Card()
        {
                name = "";
        }
        public Card(String n)
        {
                name = n;
        }
        public String getName()
        {
                return name;
        }
        public String toString()
        {
                return "Cardholder: " + getName();
        }
}
```

[10 marks]

(c) What is meant by the term Polymorphism? [3 marks]

(d) Write a program that creates an ArrayList that contains both types of card and then iterates over the list to output the toString() method. [10 marks]

Q3. (a) Show the code required to allow the user to enter a string and store that string in a text file. [10 marks]

(b) Describe how you would write an object to a file. [10 marks]

(c) Show the code required to write an object to a file. [10 marks]

Q4. (a) Please answer true or false to the following questions:

1. A try block must be followed by a catch *and* a finally block.
2. If you write a method that might cause a compiler-checked exception, you *must* wrap that risky code in a try *I* catch block.
3. Catch blocks can be polymorphic.
4. Only 'compiler checked' exceptions can be caught
5. If you define a try / catch block, a matching finally block is optional.
6. If you define a try block, you can pair it with a matching catch or finally block, or both.
7. If you write a method that declares that it can throw a compiler-checked exception, you must also wrap the exception throwing code in a try / catch block.
8. The main ( ) method in your program must handle all unhandled exceptions thrown to it.
9. A single try block can have many different catch blocks.
10. A method can only throw one kind of exception.
11. A finally block will run regardless of whether an exception is thrown.
12. A finally block can exist without a try block.
13. A try block can exist by itself, without a catch block or a finally block.
14. The order of catch blocks never matters.
15. A method with a try block and a finally block, can optionally declare the exception.
16. Runtime exceptions must be *handled* or *declared*.

[8 marks]

(b) Will the following code run and if not what fixes do you need to make? [6 marks]

(c) What is the final output? [2 marks]

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

public class InnerButton
{
   JFrame frame;
   JButton b;

   public static void main(String[]args)
   {
      InnerButtan qui = new InnerButton();
      gui.go () ;
   }

   public void go()
   {
      frame = new JFrame();
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      b = new JButton("A");
      b.addActionListener();

      frame.getContentPane().add(BorderLayout.SOUTH, b);
      frame.setSize(200,lOO);
      frame.setVisible(true);
   }

   class BListener extends ActionListener
   {
      public void actionPerformed(ActionEvent e)
      {
         if (b.getText().equals("A"))
         {
            b.setText("B");
         } else {
            b. setText ("A");
         }
      }
   }
}
```

(d) Explain the following code which is part of a Database manager class. What errors may occur when executing the code? [6 marks]

```
public class BooksRepository {

  private static BooksRepository instance;
  private static final String user="deitel" ;
  private static final String password="deitel";
  private static final String DATABASE_URL = "jdbc:mysql://localhost/books";

  private Connection connection = null;
  private PreparedStatement statement;

  public static BooksRepository getInstance() throws SQLException
   {
        if (instance == null)
             instance = new BooksRepository();
        return instance;
   }



  private BooksRepository () throws SQLException
   {
        String getAuthors=  "SELECT authorID, firstName, lastName FROM authors";
        try {
           connection = DriverManager.getConnection(  DATABASE_URL, user, password );
           statement  = connection.prepareStatement(getAuthors);

        }catch (SQLException se) {
          Se.printStackTrace(););
           }
  }
 void outputAuthors() throws SQLException
 {… }
 public void destroy () {…}
 }
```

(e) Complete the methods outputAuthors() and destroy() [8 Marks]