Appendices Attached

First year overseas students are allowed the use of a pre-approved simple English language translation, non-scientific dictionary.

Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Object Oriented Programming 1

Date: Thursday 22\textsuperscript{nd} May 2008

Time: 14:00 – 16:00

\textit{QUESTION PAPER MUST NOT BE REMOVED FROM THE EXAM ROOM}

Please answer ALL Questions in Section A
Answer ONE Question in Section B and ONE Question in Section C

Use a separate answerbook for EACH section

The use of electronic calculators is NOT permitted.
Section A

Answer ALL Questions

A1. Briefly explain whether each of the following statements will compile in Java:

```java
int score = 3.5;
double aNum = 2000;
```

(2 marks)

A2. What is the effect of the modifiers `static` and `final` in the following declaration?

```java
public static final int PASS_MARK = 55;
```

(2 marks)

A3. Write two Java expressions that will evaluate the division of the `int` value `a` by the `int` value `b`, and the remainder.

(2 marks)

A4. A DirectoryEntry class is defined as follows:

```java
public class DirectoryEntry {
    String firstName, surname;
    String address;
    int areaCode;
    long number
}
```

Write a `toString()` method to return this information formatted as you would expect to find it in a phone book.

(2 marks)
A5. Employees pay National Insurance (NI) at the rate of 10% on all their income, but only if their monthly income exceeds £300.00; otherwise they pay no NI. Write a Java if statement to calculate \( ni \) for any given amount of monthly \( pay \). The variables \( ni \) and \( pay \) are both declared as double.

(2 marks)

A6. Write the body of a Java static method, with the signature

\[
\text{public static double min(double a, double b)}
\]

that will return whichever of the two double arguments has the smaller value. (2 marks)

A7. Declare a character array, \textit{vowels} holding the values \textit{a}, \textit{e}, \textit{i}, \textit{o}, and \textit{u}. (2 marks)

A8. What \textit{type} of data will be the result of the expression ""+456.7? Why is it of that type? (2 marks)

A9. In a savings account, savers deposit a sum of money (\textit{principal}) and at the end of each whole year, the current value of principal is multiplied by the \textit{interestFactor}. (An \textit{interestFactor} of 1.05 would correspond to an interest rate of 5%).

Write a Java \textit{for} loop to predict what \textit{principal} will be after \textit{y} years with a given \textit{interestFactor}. (2 marks)

A10. The \textit{String} variable \textit{weightIn} contains a string that should represent a real number. Write a Java statement to assign that real value to the \textit{double} variable \textit{weight}, ignoring the need to check for a malformed number. (2 marks)

[PTO]
A11. What is meant by re-throwing an exception? Write a short Java fragment to illustrate this. (3 marks)

A12. What is meant by claiming an exception? Why is it sometimes better to do this than to catch the exception? (2 marks)

A13. A point-of-sale system stores its inventory in a HashMap of StockItem objects, each of which has a unique `itemCode` (int), a `description` (String), `quantity` (int) and a `unitPrice` (double). All the fields of a StockItem are declared private. Write a constructor for the StockItem class. (3 marks)

A14. What is the purpose of a class implementing the Comparable interface? (3 marks)

A15. Consider the following Java collections classes: (1) ArrayList, (2) LinkedList, (3) HashSet, (4) TreeSet, (5) HashMap and (6) TreeMap.

i) Of all the types of collection mentioned above, which would be the most appropriate to store a collection of library loan records indexed by the accession number of the book, and why? (3 marks)

ii) Which would be the most appropriate collection class for storing the distinct words used in a text, and why? (3 marks)

iii) Which would be the most appropriate type of collection for storing the collection of individual OrderItem objects that belong to a PurchaseOrder, and why? (3 marks)
Section B

Answer ONE question in this Section

B1. a) A class called Course is required by a programmer who is developing a system for use in an adult education college. A Course object will require the following attributes: a course identity number and name (both of which will be strings), and the number of hours the course operates in a week. It will also need an attribute to record the hourly fee - this will be the same for every object of the class Course.

The identity number, name and hours per week will need to be set at the time a Course object is created; the first two will not need to be changed after this. However, it will be necessary to provide a method that allows the number of hours per week to be re-set during the object's lifetime. In addition, class methods are required for the purposes of setting and getting the hourly fee. Finally, methods are required to read the values of the remaining attributes, as well as a method that calculates and returns the total weekly cost of the course.

i) Write Java code for the class Course. (8 marks)

ii) Assume the following variables have been declared within a program:

   String id;
   String name;
   int hoursPerWeek;

   Now write a fragment of code (not a whole program) that uses the Course class and obtains the course identity number, name and number of hours per week from a user and creates a new Course object. (4 marks)

iii) Write a line of code that will output (on a console screen) the total weekly cost of the course for the Course object defined in (ii). (2 marks)

b) What is a wrapper class? Give an example of such a class. (2 marks)

c) Explain the meaning of the reserved words static and final. (4 marks)
B2. a) Explain what is meant by a **layout manager** in Java. (4 marks)

b) The figure below shows an application that allows a user to enter a length in feet and inches and then press a button which converts this length to metres displayed to the nearest centimetre. The screenshot shows the result of pressing the “Calculate” button after inputting a length of 5 feet 11 inches.

![Imperial to Metric Calculator](image)

i) Copy the figure to your answer booklet and label all the Swing components used in the construction of the graphical user interface. (4 marks)

ii) The code for the interface is implemented in a class called `ConverterGUI` which extends the `JPanel` class. Write the Java code that declares the graphics components and implements a constructor method for this class which adds these components to the `JPanel`. Your code should ensure that the components do not get moved around when the window is arbitrarily resized. (6 marks)

iii) Assuming that the GUI class implements the `ActionListener` interface, write the Java code for the `actionPerformed` method which enables the application to respond to a mouse click event. You are not required to include validation code for the input fields.

(You may find the following information useful: 1 inch = 2.54cm, 12 inches = 1 foot.). (6 marks)
C1. Consider the application described in Appendix 1, and answer the following questions.

A Card object is constructed using the Enum types described in Appendix 1 as follows:

```java
Card c1 = new Card(Suit.CLUBS, Value.TWO);
```

a) Looking at the class definitions in Appendix 1, answer the following question: How would you write a statement to construct a Player object?

i) `Player aPlayer = new Player("John Smith");`
ii) `Player aPlayer = new Player(c1);`
iii) `Player aPlayer = new Player(new String("c1"));`
iv) `Player aPlayer = new Player("John Smith", c1);` (3 marks)

b) Looking at the class definitions in Appendix 1, answer the following question: How would you write a statement to add a card into a player’s hand?

i) `boolean success = aPlayer.newCard("c1");`
ii) `c1 = aPlayer.newCard();`
iii) `c1.newCard(aPlayer);`
iv) `aPlayer.newCard(c1);` (3 marks)

Assume that a player named John Smith is playing cards and he has currently in his hand the following cards (in this order):

Card `c1 = new Card(Suit.CLUBS, Value.TWO);`
Card `c2 = new Card(Suit.CLUBS, Value.THREE);`
Card `c3 = new Card(Suit.CLUBS, Value.FOUR);`
Card `c4 = new Card(Suit.CLUBS, Value.FIVE);`
Card `c5 = new Card(Suit.CLUBS, Value.SIX);`

Assume that a player named John Smith is playing cards and he has currently in his hand the following cards (in this order):

Card `c1 = new Card(Suit.CLUBS, Value.TWO);`
Card `c2 = new Card(Suit.CLUBS, Value.THREE);`
Card `c3 = new Card(Suit.CLUBS, Value.FOUR);`
Card `c4 = new Card(Suit.CLUBS, Value.FIVE);`
Card `c5 = new Card(Suit.CLUBS, Value.SIX);`

c) Which output would method `toString()` of class `Player` produce when called on the `Player` object that represents John Smith?

i) `(John Smith: [2C, 3C, 4C, 5C, 6C])`
ii) `John Smith: [2C, 3C, 4C, 5C, 6C]`
iii) "John Smith": [2C, 3C, 4C, 5C, 6C]
iv) "John Smith": [3C, 4C, 2C, 5C, 6C] (3 marks)

(Question C1 continues on the following page)
d) Which of the classes and enum types listed below have their `toString()` method called at some point in the execution of method `toString()` of class `Player`, when this method is called?


i) Only (1) and (2).
ii) Only (2).
iii) (1), (2), (3), (4) and (5).
iv) Only (1) and (3). (2 marks)

e) If you decide to modify the type of data member `hand` of class `Player` to a `HashSet` instead of an `ArrayList`, which of the following methods MUST be changed due to the use of the `HashSet`?

i) Only the constructor of class `Player`.
ii) Only method `newCard` of class `Player`.
iii) All the methods of class `Player`.
iv) None of the methods of class `Player`. (3 marks)

Consider the application described below, and answer the following question.

```java
public abstract class Animal {
    private int yearOfBirth;
    private String specie;
    private final int CURRENTYEAR = 2008;

    public Animal(int yearOfBirth, String specie) {
        if (yearOfBirth <= CURRENTYEAR)
            this.yearOfBirth = yearOfBirth;
        this.specie = specie;
    }

    public int age() {
        return CURRENTYEAR - this.yearOfBirth;
    }

    public abstract String gender();
}
```

(Question C1 continues on the following page)
(Question C1 continues from the previous page)

f) Which of the following definitions of class Female will NOT cause a compiler error?

(i)
```java
public class Female extends Animal {
    private int numOfChildren;
    public Female(int yearOfBirth, String species, int numOfChildren)
    {
        super(yearOfBirth, species);
        this.numOfChildren = numOfChildren;
    }
    public int getNumOfChildren()
    {
        return this.numOfChildren;
    }
    public String gender()
    {
        return "Female";
    }
}
```

(ii)
```java
public class Female extends Animal {
    private int numOfChildren;
    public Female(int yearOfBirth, String species, int numOfChildren)
    {
        super(yearOfBirth, species);
        this.numOfChildren = numOfChildren;
    }
    public int getNumOfChildren()
    {
        return this.numOfChildren;
    }
}
```
(iii)  
```
public class Female extends Animal {
    private int numOfChildren;
    public Female(int yearOfBirth, String species, int numOfChildren) {
        super(yearOfBirth, species);
        this.numOfChildren = numOfChildren;
    }
    public int getNumOfChildren() {
        return this.numOfChildren;
    }
    public void setNumOfChildren(int numOfChildren) {
        this.numOfChildren = numOfChildren;
    }
    public void gender(String) {
        return "Female";
    }
}
```

(iv)  
```
public class Female extends Animal {
    private int numOfChildren;
    public Female(int yearOfBirth, String species, int numOfChildren) {
        super(yearOfBirth, species);
        this.numOfChildren = numOfChildren;
    }
    public void setNumOfChildren(int numOfChildren) {
        this.numOfChildren = numOfChildren;
    }
    public int gender() {
        return "Female";
    }
}
```

(3 marks)

(Question C1 continues on the following page)
Consider the application described below, and answer the following question.

```java
public class TestAProgram {
    public static void main(String[] args) {
        InputStreamReader input = new InputStreamReader(System.in);
        BufferedReader reader = new BufferedReader(input);
        int num = -1;
        char[] collectionOfVowels = {'a', 'e', 'i', 'o', 'u'};
        try {
            System.out.print("Enter a vowel number (1-5): ");
            num = Integer.parseInt(reader.readLine());
        }
        catch(IOException e) {
            e.printStackTrace();
        }
        if (num > 0)
            System.out.println(collectionOfVowels[num-1]);
    }
}
```

g) Which of the following statements is true of the above program?

i) This program will not compile.
ii) This program will never throw an unchecked exception.
iii) This program will compile but not run.
iv) This program may throw an IOException. (3 marks)
C2. Consider the application described in Appendix 2, and answer the following questions.

a) To insert a pair (word, meaning) into a WordDictionary, one would need to call the following methods in the following order:

i) The constructor of class Pair, the constructor of class WordDictionary, and method `addNewWordAndMeaning` of class WordDictionary.
iii) The constructor of class WordDictionary, method `addNewWordAndMeaning` of class WordDictionary, and the constructor of class Pair.  (3 marks)

b) Suppose you have a Pair object in your program. To check the meaning of the word in the pair, the following methods can be called, in the following order, except:

iii) Method `equals` of class Pair, and `toString` method of class String.
iv) Method `getMeaning` of class Pair, and the `toString` method of class String.  (3 marks)

c) Suppose you want to simplify the implementation of the WordDictionary application by using only class WordDictionary, instead of two classes (Pair and WordDictionary). To achieve this, you could do the following:

i) Simply remove class Pair from the application, leaving class WordDictionary as it is.
ii) Remove class Pair from the application and change the implementation of class WordDictionary to use a HashMap<String, String>, and implement/modify all the necessary methods, making use of the methods of class HashMap.
iii) Remove class Pair from the application and change the implementation of class WordDictionary to use an ArrayList<String>, and implement/modify the current methods of the class to deal with the ArrayList<String> where necessary.

(Question C2 continues on the following page)
iv) Remove class Pair from the application and change the implementation of class WordDictionary to use a HashSet<String, String>, and implement/modify the current methods of the class to deal with the HashSet<String, String> where necessary. (3 marks)

d) If you want to make the collection of pairs (word, meaning) persistent, i.e. to be able to save the dictionary to disk every time you close the application, and to be able to read it back into memory when running the application again, what can you do?

i) Only write an import statement to import package java.io on top of file containing class WordDictionary.

ii) Import the proper package and define two more methods in class WordDictionary, one to read a WordDictionary object from disk and another to write a WordDictionary to disk.

iii) Import the proper package and create a FileHandler class, defining methods to read a WordDictionary object from and write a WordDictionary to disk.

iv) Two of the above. (3 marks)

e) When writing methods for reading data from and writing data to disk, it is mandatory to use the following:

i) Static methods to read data from and write data to disk, as well as import the java.util package.

ii) Static methods in a file handler class to be called from elsewhere in the application.

iii) A .txt file to store all of the data in disk.

iv) Try and catch blocks to enclose the relevant code, to catch the IO Exceptions that may occur during file writing and reading. (3 marks)
Consider the application described below, and answer the following question.

```java
public class StandardLetter implements Letter {
    private String author;
    private boolean checked;
    public StandardLetter(String author) {
        this.author = author;
    }
    public void send(String destination) {} {
    public void receive(String origin) {
        if (origin.length() > 0)
            checked = true;
        else checked = false;
    }
}

public class ReferenceLetter implements Letter {
    private String onBehalfOf;
    private boolean checked;
    public ReferenceLetter(String onBehalfOf) {
        this.onBehalfOf = onBehalfOf;
    }
    public void send(String destination) {
        if (destination.length() > 0)
            checked = true;
        else checked = false;
    }
    public void receive(String origin) {} {
}

public class TestApplication {
    public static void main(String[] args) {
        StandardLetter letter1 = new StandardLetter("John Smith");
        ReferenceLetter letter2 = new ReferenceLetter("Maria Castro");
        letter1.send("USA");
        letter2.send("Spain");
    }
}
```

(Question C2 continues on the following page)
f) Which of the following statements is true of the above program?

i) Letter will not compile successfully.
ii) StandardLetter will not compile successfully.
iii) ReferenceLetter will not compile successfully.
iv) All classes in the application will compile successfully. (3 marks)

Consider the application described below, and answer the following question.

```java
public class TestAProgram {
    public static void main(String[] args) {
        InputStreamReader input = new InputStreamReader(System.in);
        BufferedReader reader = new BufferedReader(input);
        int num = -1;
        char[] collectionOfVowels = {'a', 'e', 'i', 'o', 'u'};
        try {
            System.out.print("Enter a vowel number (1-5): ");
            num = Integer.parseInt(reader.readLine());
        }
        catch(IOException e) {
            e.printStackTrace();
        }
        if (num > 0)
            System.out.println(collectionOfVowels[num-1]);
    }
}
```

g) Which of the following instructions from the program in the question above might throw an exception?

i) char [] vowel = {'a','e','i','o','u'};
ii) System.out.print("enter a vowel number (1-5): ");
iii) System.out.println(vowel[num-1]); (2 marks)
Appendix 1

```java
public enum Suit {
    CLUBS { public String toString(){return "C";} },
    DIAMONDS { public String toString(){return "D";} },
    HEARTS { public String toString(){return "H";} },
    SPADES { public String toString(){return "S";} }
}

public enum Value {
    TWO { public String toString(){return "2";} },
    THREE { public String toString(){return "3";} },
    FOUR { public String toString(){return "4";} },
    FIVE { public String toString(){return "5";} },
    SIX { public String toString(){return "6";} },
    SEVEN { public String toString(){return "7";} },
    EIGHT { public String toString(){return "8";} },
    NINE { public String toString(){return "9";} },
    TEN { public String toString(){return "10";} },
    JACK { public String toString(){return "J";} },
    QUEEN { public String toString(){return "Q";} },
    KING { public String toString(){return "K";} },
    ACE { public String toString(){return "A";} }
}

public class Card {
    private final Suit suit;
    private final Value value;

    public Card(final Suit suit, final Value value) {
        this.suit = suit;
        this.value = value;
    }

    public String toString() {
        return value.toString() + suit.toString();
    }
}

public class Player {
    private String name;
    private ArrayList<Card> hand = new ArrayList<Card>();

    public Player(final String name) {
        this.name = name;
    }

    public void newCard(final Card card) {
        hand.add(card);
    }

    public String toString() {
        String result = new String(this.name + ': ');
        return result + this.hand.toString();
    }
}
```
public class Pair {
    private String word;
    private String meaning;
    public Pair(String word, String meaning) {
        this.word = word;
        this.meaning = meaning;
    }
    public String getWord() {
        return this.word;
    }
    public String getMeaning() {
        return this.meaning;
    }
    public boolean equals(Pair pair) {
        if (this.word.equals(pair.getWord()) && 
            (this.meaning.equals(pair.getMeaning())))
        {
            return true;
        }
        return false;
    }
    public String toString() {
        return this.word.toString() + ": " + this.meaning.toString();
    }
}

public class WordDictionary {
    private ArrayList<Pair> dict;
    public WordDictionary() {
        this.dict = new ArrayList<Pair>();
    }
    public boolean addNewWordAndMeaning(Pair wordAndMeaning) {
        return this.dict.add(wordAndMeaning);
    }
    public boolean removeWordAndMeaning(Pair wordAndMeaning) {
        return this.dict.remove(wordAndMeaning);
    }
    public boolean contains(Pair wordAndMeaning) {
        if (this.dict.contains(wordAndMeaning))
            return true;
        else return false;
    }
    public boolean contains(String word) {
        boolean result = false;
        for (int pos = 1; pos < dict.size(); pos++)
        {
            if (this.dict.get(pos).getWord().equals(word))
            {
                result = true;
            }
        }
        return result;
    }
}
}  
    return result;
  }
public String getMeaningFromWord(String word) {
  String result = null;
  for (int pos = 1; pos < this.dict.size(); pos++) {
    if (this.dict.get(pos).getWord().equals(word)) {
      result = this.dict.get(pos).getMeaning();
    }
  }
  return result;
}
public String toString() {
  return this.dict.toString();
}
}