University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

SEMESTER I EXAMINATION – 2012/2013

COMP 20010

Data Structures & Algorithms I

Prof. A. Mille

Mr. J. Dunnion

Dr Eleni Mangina

Time allowed: 2 hours

Instructions for candidates

Answer **any** two questions. All questions carry equal marks (50).

Use of calculators is prohibited.

Instructions for invigilators

Use of calculators is prohibited.

1. Answer parts (a) to (d). 50 marks in total

(a) (15 marks)

(i) Briefly define the following variable modifiers in terms of scope (or visibility) of instance variables: public, protected, private, final. **(4 marks)**

(ii) What is the output of the following program? Why? **(5 marks)**

```
public class AverageCalculator
{
            public static void main(String[] args)
             {
               int age1 = 18;
                int age2 = 35;
                int age3 = 50;
                int age4 = 44;
                double averageAge = (age1 + age2 + age3 + age4) / 4;
                System.out.println(averageAge);
              }
}
```

(iii) An int variable is four bytes in length. In the wrapper class Integer, there is a constant called Integer.SIZE which represents the number of bits the variable contains. The program below prints this constant. Modify the program so that it prints the sizes of each of the primitive numeric types. Try to include all eight primitive types. One class does not have a SIZE constant. Which class is it?

(6 marks)

```
public class PrimitiveSizes
{
            public static void main(String[] args)
            {
             System.out.println("An int variable is " + Integer.SIZE + " bits.");
             }
}
```

(b) (15 marks)

(i) Write a program that reads in the names and scores of students and then computes and displays the names of the students with the highest and lowest scores. **(5 marks)**

A simple method of carrying out this task would be to have two parallel arrays:

```
String[] names;

int[] scores;
```

However, you should avoid parallel arrays in your solution.
First, write a class `Student` to solve the parallel array problem. A student should have a name and a score.

(ii) Listed below is a bad implementation of the `StudentScores` class that uses two parallel arrays. Modify it to eliminate the use of parallel arrays. Use an array list of students in your solution. Your improved `StudentScores` class should use the `Student` class that you created previously.**(10 marks)**

```
public class BadStudentScores
{
   private final int MAX_STUDENTS = 100;
   private String[] names;
   private int[] scores;
   private int numStudents;

   public BadStudentScores()
   {
      scores = new int[MAX_STUDENTS];
      names = new String[MAX_STUDENTS];
      numStudents = 0;
   }

   public void add(String name, int score)
   {
      if (numStudents >= MAX_STUDENTS)
         return; // Not enough space to add new student score
      names[numStudents] = name;
      scores[numStudents] = score;
      numStudents++;
   }

   public String getHighest()
   {
      if (numStudents == 0)
         return null;
      int highest = 0;
      for (int i = 1; i < numStudents; i++)
         if (scores[i] > scores[highest])
            highest = i;
      return names[highest];
   }

   public String getLowest()
   {
      if (numStudents == 0)
         return null;
      int lowest = 0;
      for (int i = 1; i < numStudents; i++)
         if (scores[i] < scores[lowest])
            lowest = i;
      return names[lowest];
   }
}
```

(c) We often need to convert a value from a `String` to a primitive data type. For example, the program below converts the `String` `"12345"` to an `int` using the static method `parseInt` that is contained in the wrapper class `Integer`. Complete the program below by filling in the code specified in the embedded comments. **(10 marks)**

```
public class StringConversion
{
  public static void main(String[] args)
```

```
    {
      String value1 = "12345";
      int intValue = Integer.parseInt(value1);
      System.out.println("intValue = " + intValue);

      String value2 = "12.345";
      // Convert value2 to a double here
      // Print the converted value

      String value3 = "87654";
      // Convert value3 to a long here
      // Print the converted value

      String value4 = "321";
      // Convert value4 to a short here
      // Print the converted value

      String value5 = "-28";
      // Convert value5 to a byte here
      // Print the converted value

      String value6 = "6";
     // Convert value6 to a char here.  (Hint: See Advanced Topic 3.5)
      // Print the converted value

      String value7 = "true";
      // Convert value7 to a boolean here.  (Hint: Check the API for the
      // Boolean wrapper class.  Which method returns a Boolean?)
      // Print the converted value

      String value8 = "-45.237";
      // Convert value7 to a float here
      // Print the converted value
    }
  }
```

(d) (10 marks)

(i) Which values of `year` cause the following loop to terminate with a correct answer? **(5 marks)**

```
/**
  Counts the number of years from a year input by the user
  until the year 3000.
*/
public class CountYears
{
  public static void main(String[] args)
  {
    int millennium = 3000;
    Scanner in = new Scanner(System.in);
    System.out.print("Please enter the current year: ");

    int year = in.nextInt();
    int nyear = year;

    while (nyear != millennium)
    {
      nyear++;
      }

      System.out.println("Another " + (nyear - year) + " years to the millennium.");
   }
}
```

(ii) Convert this `while` loop to a `for` loop **(5 marks)**

```
/**
 Program to compute the first integral power to which 2 can be
raised that is greater than that multiple of a given integer.
*/
public class CountPowerOf2
{
   public static void main(String[] args)
   {
      Scanner in = new Scanner(System.in);
      System.out.print("Please enter a number, 0 to quit: ");
      int n = in.nextInt();
      int i = 1;
                while (n * n > Math.pow(2, i))
               {
                        i++;
                }
      System.out.println("2 raised to " + i
      + " is the first power of two greater than " + n + " squared");
               }
 }
```

2. Answer parts (a) to (d). **50 marks in total**

(a) (5 marks)

Consider the following problem: A company allows its employees to check out certain items, such as handheld computers and music players, to gain personal experience with them. Popular items can be reserved on a first come/first served basis. A reservation list is kept for each item. There is a fine for overdue items. Your development team's task is to write a software program that allows the stockroom clerk to check out items and check them back in, reserve items, notify employees when a reserved item has been returned, produce reports of overdue items, and track payment of fines. Based on the rule of thumb for finding classes, what classes would you choose to implement this program?

(b) (15 marks)

Briefly define the following terms that are commonly used while designing classes and provide a simple example for each term: Coupling between classes, Immutable Classes, Side effect of a method, Static method, Static Variable

(c) (10 marks)

The following class has a method with a **side effect**:

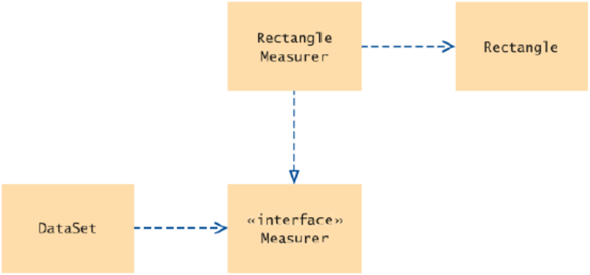
```
/** A purse computes the total value of a collection of coins.*/
public class Purse
{            private double total;
              /**
             Constructs an empty purse.
             */
             public Purse()
             {
                          total = 0;
             }
              /**
             Adds a coin to the purse.
             @param aCoin the coin to add
             */
             public void add(Coin aCoin)
             {
                          total = total + aCoin.getValue();
                          System.out.println("The total is now " + total);
             }
              /**
             Gets the total value of the coins in the purse.
             @return the sum of all coin values
               */
               public double getTotal()
               {
                            return total;
               }
}
```

(i) Describe the side effect and explain why it is not desirable.

(ii) How would you eliminate the side effect in the `add` method?

(d) (20 marks)

(i) Define the interface types in Java and denote the differences with classes.
(ii) Provide the general syntax of implementing an interface in Java and a simple paradigm.
(iii) Define the terms polymorphism and inner class in Java.
(iv) Given the following UML diagram comment on the types of classes and their relationships.



3. Answer parts (a) to (c). 50 marks in total

(a) (20 marks)

(i) A technique that can be used to describe the running time of an algorithm is known as *asymptotic notation*. One form of this technique is known as “Big-Oh” notation. Give the definition for this notation. **(3 marks)**

(ii) In the following, use either a direct proof (by giving values for c and n_0 in the definition of “Big-Oh” notation) or cite one of the rules given in the textbook or in the lecture slides:
Show that if f(n) is O(g(n)) and d(n) is O(h(n)), then f(n)+d(n) is O(g(n)+h(n)). **(5 marks)**

(iii) In the following, use either a direct proof (by giving values for c and n_0 in the definition of big-Oh notation) or cite one of the rules given in the textbook or in the lecture slides:
Show that $3(n+1)^7 + 2n\log n$ is $O(n^7)$. **(5 marks)**

(iv) Order the following functions by asymptotic growth rate: **(3 marks)**

$4n\log n + 2n$
$3n + 100\log n$
$n^2 + 10n$

2^{10}
$4n$
n^3

$2^{\log n}$
2^n
$n\log n$

*(v)*Show that if d(n) is O(f(n)) and e(n) is O(g(n)), then the product d(n)e(n) is O(f(n)g(n)). **(4 marks)**

(b) (15 marks)

(i) What is *pseudo code*? List the set of programming constructs that are typically associated with pseudo code. **(5 marks)**

(ii) What does the following algorithm do? Analyze its worst-case running time and express it using the “Big-Oh” notation. **(4 marks)**

```
Algorithm Foo (a, n):
    Input: two integers, a and n
    Output: ?
k ← 0
b ← 1
while k < n  do
    k ← k + 1
    b ← b * a
return  b
```

(iii) What does the following algorithm do? Analyze its worst-case running time and express it using the “Big-Oh” notation. **(6 marks)**

```
Algorithm Bar (a, n):
    Input: two integers, a and n
    Output: ?
```

```
k ← n
b ← 1
c ← a
while k > 0  do
    if k mod 2 = 0 then
            k ← k/2
            c ← c * c
    else
            k ← k – 1
            b ← b*c
return  b
```

(c) (15 marks)

Bill has an algorithm, find2D, to find an element x in an n x n array A. The algorithm find2D iterates over the rows of A, and calls the algorithm arrayFind, on each row, until x is found or it has searched all rows of A. What is the worst-case running time of find2D in terms of n? What is the worst-case running time of find2D in terms of N, where N is the total size of A? Would it be correct to say that find2D is a linear-time algorithm? Why or why not?

```
Algorithm arrayFind(x,A)
        Input: An element x and an n-element array, A
        Output: The index i such that x = A[i] or
                -1 if no element of A is equal to x
        i ←0
        while i < n do
                if x = A[i] then
                        return i
                else
                        i ← i + 1
        return - 1
```

4. Answer parts (a) to (c). **50 marks in total**

(a) **(15 marks)**

(i) What is an ArrayList? List the methods commonly associated with the ArrayList. **(5 marks)**

(ii) Briefly define the following terms: Wrapper class, autoboxing and enhanced "for" loop and provide examples for each one. **(10 marks)**

(b) **(15 marks)**

(i) What is a *Stack*? List the operations commonly associated with the Stack Abstract Data Type (ADT). **(5 marks)**

(ii) Describe the output of the following series of stack operations on a single, initially empty stack: *push(5), push(3), pop(), push(2), push(8), pop(), pop(), push(9), push(1), pop(), push(7), push(6), pop(), pop(), push(4), pop(), pop().* **(5 marks)**

(iii) Describe how to implement two stacks using one array. The total number of elements in both stacks is limited by the array length; all stack operations should run in O(1) time. **(5 marks)**

(c) **(20 marks)**

(i) What is a *Queue*? List, giving a brief description for each one, the operations commonly associated with the Queue Abstract Data Type (ADT). **(5 marks).**

(ii) What is a *Deque*? Give the pseudo-code algorithm for insertion into the front of a Deque. **(5 marks)**

(iii) Describe in pseudo code a linear-time algorithm for reversing a queue Q. To access the queue, you are only allowed to use the methods of queue ADT. *Hint: Consider using a Stack.* **(5 marks)**

(iv) Another variant of the basic Queue Abstract Data Type is a *Priority Queue*. Explain how this Priority Queue's work. **(5 marks)**