

## CSC1016S Assignment 7

Arrays and Inheritance in Java

### Assignment Instructions

This assignment concerns (i) the use of class or 'static' variables and methods, (ii) the use of arrays, and the basics of inheritance in Java question.

Some concepts from Assignment 6 are still very relevant in this assignment. This assignment is a continuation of the previous assignment on References.

One use of class variables and methods is for defining collections of constants and routines – much like a Python module. The class "`java.lang.Math`" is an excellent example. It defines constants such as  $\pi$ , and provides routines for calculating square roots, powers etc. We don't create a 'Math' object to make use of these facilities.

### Question 1 [35 marks]

This question concerns the construction of a `NumberUtils` class declaration that contains a collection of useful routines.

Write a class declaration that satisfies the following specification:

#### Class `NumberUtils`

The `NumberUtils` class contains a collection of routines for working with integers.

#### Instance variables

*None*

#### Constructors

```
private NumberUtils() {}  
    // A private, empty-bodied constructor prevents NumberUtil objects from being created.
```

#### Methods

```
public static int[] toArray(int number)  
    // Given a number that is n digits in length, maps the digits to an array length n.  
    // e.g. given the number 5678, the result is the array {5, 6, 7, 8}.
```

```
public static int countMatches(int numberA, int numberB)  
    // Given two numbers, count the quantity of matching digits – those with the same value and  
    // position. For example, given 39628 and 79324, there are 2 digits in common: x9xx2x.  
    // It is assumed that the numbers are the same length and have no repeating digits.
```

```
public static int countIntersect(int numberA, int numberB)  
    // Count the quantity of digits that two numbers have in common, regardless of position.  
    // For example, given 39628 and 97324, there are 3 digits in common: 3, 9, 2.  
    // It is assumed that the numbers are the same length and have no repeating digits.
```

You should make a simple test program (which you do not need to submit) to check your code.

## Question 2 [35 marks]

This question concerns the construction of a simple game called 'Cows and Bulls' that is implemented using the NumberUtils class of question one.

The game involves guessing a mystery 4 digit number. The number contains no repeat digits and no zero.

With each guess, the number of correct digits – bulls – and the number of digits that would be correct if they were in the right position – cows.

For example, say the mystery number is 8657 and the player guesses 4678, there is one bull (6), and two cows (7, 8).

### Class CowsAndBulls

CowsAndBulls implements the logic for a cows and bulls guessing game the player has

#### Constants

```
public final static int NUM_DIGITS = 4;
public final static int MAX_VALUE = 9876;
public final static int MIN_VALUE = 1234;
public final static int MAX_GUESSES = 10;
```

#### Constructors

```
public CowsAndBulls(int seed)
    // Create a CowsAndBulls game using the given randomisation seed value to generate
    // a mystery number of NUM_DIGITS length, and that gives the player MAX_GUESSES guesses.
```

#### Methods

```
public int guessesRemaining()
    // Obtain the number of guesses remaining.

public Result guess(int guessNumber)
    // Evaluates a guess that the mystery number is guessNumber, returning the outcome in the form
    // of a Result object. Decrements guesses remaining.
    // Assumes that game is not over.

public int giveUp()
    // End the game, returning the secretNumber.

public boolean gameOver()
    // Returns true if (i) the secret number has been guessed, or (ii) there are no more guesses.
```

On the Amathuba web page you will find:

- A NumberPicker class that you should use in the CowsAndBulls constructor to generate a mystery number,
- A Result class that you should use to implement the CowsAndBulls guess method.
- A Game program that completes the game; implementing user interaction.

NOTE:

- the seed value supplied to the CowsAndBulls constructor should be fed to the NumberPicker object that it creates. The purpose of this value is to make the (pseudo) random number generation repeatable i.e. the same seed value causes the same mystery number to be generated.
- To generate a mystery number, create a NumberPicker with the seed value and a range 1-9 (inclusive). Get a digit, multiply by 10 and add a digit, multiply by 10 and add a digit,...

Running the Game program produces the following sorts of interaction.

**Sample IO** (The input from the user is shown in **bold font** – do not program this):

```
Your challenge is to guess a secret 4 digit number.
Enter randomisation seed value:
10
Make a guess:
5678
Sorry that's incorrect.
You have 3 cows and 1 bull.
You have 9 guesses remaining
Make a guess:
5687
Sorry that's incorrect.
You have 2 cows and 2 bulls.
You have 8 guesses remaining
Make a guess:
8657
Correct !
```

**Sample IO** (The input from the user is shown in **bold font** – do not program this):

```
Your challenge is to guess a secret 4 digit number.
Enter randomisation seed value:
5
Make a guess:
1234
Sorry that's incorrect.
You have 0 cows and 2 bulls.
You have 9 guesses remaining
Make a guess:
4569
Sorry that's incorrect.
You have 2 cows and 0 bulls.
You have 1 guesses remaining
Make a guess:
4537
Sorry, you lose.
```

### Question 3 [30 marks]

Write a program to demonstrate the use of inheritance by creating and outputting 3 simple objects, where the classes for the second and third objects inherit from the class for the first object.

The first object is a Shape with a name and colour. The second object is a Rectangle with additional length and width. The third object is a Circle with additional radius.

Use the provided *Question3.java* file. Use constructors and override methods as appropriate.

**Sample IO** (The input from the user is shown in **bold** font – do not program this):

```
Pentagon Blue
Circle Purple Radius 3.0
Rectangle Red Length 6.0 Width 8.0
```

### Marking and Submission

Submit `NumberUtils.java`, `CowsAndBulls.java`, `Shape.java`, `Rectangle.java`, and `Circle.java` in a single .ZIP folder to the automatic marker.

The zipped folder should have the following naming convention:

***yourstudentnumber.zip***