

The University of Melbourne  
Department of Computer Science and Software Engineering  
**433-254 Software Design**  
Semester 2, 2003  
**Answers for Lab 2**  
Week 3

## 1. Writing Simple Classes:

- a) In Java systems, there is only one class with the 'main' function, which initiates the execution of the whole system. Therefore, in order to test individual classes (i.e. unit testing), developers are usually writing a simple class with a main function (also known as a driver class) that simply calls all different functions of a target class and prints the results.

Your task is to download the file "Keyboard.java" from:

[www.cs.mu.oz.au/254/Keyboard/Keyboard.html](http://www.cs.mu.oz.au/254/Keyboard/Keyboard.html)

and write a simple driver program that tests the following methods from Keyboard.java:

readString()	readInt()
readWord()	readLong()
readChar()	readFloat()
readBoolean()	readDouble()

### Sample Answer:

```
// A class to execute one or more Keyboard methods
class Test
{
    public static void main(String[] args)
    {
        System.out.print("Please enter a string: ");
        String str = Keyboard.readString();
        System.out.println("String = " + str);
        System.out.println();

        System.out.print("Please enter a word: ");
        String word = Keyboard.readWord();
        System.out.println("word = " + word);
        System.out.println();

        System.out.print("Please enter a character: ");
        char ch = Keyboard.readChar();
        System.out.println("char = " + ch);
        System.out.println();

        System.out.print("Please enter a boolean (true/false): ");
        boolean bool = Keyboard.readBoolean();
        System.out.println("boolean = " + bool);
        System.out.println();
    }
}
```

```

        System.out.print("Please enter an int number: ");
        int numInt = Keyboard.readInt();
        System.out.println("int = " + numInt);
        System.out.println();

        System.out.print("Please enter a long number: ");
        long numLong = Keyboard.readLong();
        System.out.println("long = " + numLong);
        System.out.println();

        System.out.print("Please enter a float number: ");
        float numFloat = Keyboard.readFloat();
        System.out.println("float = " + numFloat);
        System.out.println();

        System.out.print("Please enter a double number: ");
        double numDouble = Keyboard.readDouble();
        System.out.println("double = " + numDouble);
        System.out.println();
    }
} // end class

```

- b) Write a Java class Student to meet the following specification. - The class should be able to support a 5 digit student ID, student name, marks for 3 subjects. You should have methods to set and get each of the attributes, and calculate the average for the student. Write a tester program to test your class. You should create 2 or 3 students and write code to test the class.  
 Aim - Understand how to define a class and create objects of the class.

**Sample Answer:**

```

class Student
{
    // data members
    private String _studentNumber;
    private String _studentName;
    private int _markForMaths;
    private int _markForEnglish;
    private int _markForScience;

    // Set Student Name

    public void setStudentName(String studentName)
    {
        _studentName = studentName;
    }

    // Set the student number
    public void setStudentNumber(String studentNumber)
    {
        _studentNumber = studentNumber;
    }
}

```

```

//    getNumber()
//    return student number
public String getNumber()
{
    return _studentNumber;
}

// getName()
// return student name
public String getName()
{
    return _studentName;
}

// entermarks()
// enter all subject marks given as args
public void enterMarks(int maths, int english, int science)
{
    _markForMaths = maths;
    _markForEnglish = english;
    _markForScience = science;
}

// getMathsMark()
// return mark for maths
public int getMathsMark()
{
    return _markForMaths;
}

// getEnglishMark()
// return mark for English
public int getEnglishMark()
{
    return _markForEnglish;
}

// getScienceMark()
// return mark for Science
public int getScienceMark()
{
    return _markForScience;
}

// calculateAverageMark()
// return the average of the three marks
public double calculateAverageMark()
{
    return ((_markForMaths + _markForEnglish +
            _markForScience) / 3.0);
}
}

```

```

// class StudentTester
// tests the Student class above
public class StudentTester
{
    public static void main(String[] args)
    {
        String stud_num, stud_name;

        Student s1 = new Student();
        Student s2 = new Student();

        s1.setStudentName("Britney Spears");
        s1.setStudentNumber("11111");
        s1.setStudentName("Eddie Vedder");
        s1.setStudentNumber("22222");
        s1.enterMarks(20, 60, 2);
        s2.enterMarks(80, 75, 76);

        outDetails(s1);
        outDetails(s2);
    }

    // Write the details to standard out
    public static void outDetails(Student s)
    {
        System.out.println("-----");
        System.out.println("Student Number:\t" + s.getNumber());
        System.out.println("Student Name:\t" + s.getName());
        System.out.println("Maths:\t\t" + s.getMathsMark());
        System.out.println("English:\t" + s.getEnglishMark());
        System.out.println("Science:\t" + s.getScienceMark());
        System.out.println("Average:\t" +
s.calculateAverageMark());
    }
}

```

## 2. Last Week's Extra Work:

If you haven't done the last week's extra exercises, you must do them this week! Here they are again:

- (a) Write a java program that takes your first name and last name as command line arguments to the program and displays your name and last name on separate lines. Aim: Understand the use of command line arguments.

### Sample Answer:

```

// FirstLast.java
class FirstLast
{
    public static void main(String[] args)
    {
        System.out.println("First Name : " + args[0]);
        System.out.println("Last Name : " + args[1]);
    }
}

```

- (b) Write a program that calculates the total wages based on the number of hours worked. The wages are calculated at a rate of 8.25 per hour for up to 40 hours and at the rate of 1.5 the standard rate for any hours greater than 40. Number of hours is a command line argument to the program.

Hint: Use `Integer.parseInt(String s)` converts a string to an integer (To convert the number of hours from command line to integer). We will understand this as we get through the rest of the lectures and labs but use it for now.

Aim: Understand the use of if-else and constants.

### Sample Answer:

```
// Wages.java
class Wages
{
    public static void main(String[] args)
    {
        final double RATE = 8.25;
        final int STANDARD = 40;
        double pay = 0.0;
        int hours = 0;

        hours = Integer.parseInt(args[0]);

        if (hours > STANDARD)
        {
            pay = STANDARD * RATE +
                (hours - STANDARD)*1.5*RATE;
        }
        else
        {
            pay = STANDARD * RATE;
        }
        System.out.println("Wages = " + pay);
    }
}
```

- (c) Write a program to take the student's grade as an input argument and print the comments as follows 100 (Perfect Score), 90-100 (Excellent), 80-90 (Good), 70-80 (Above Average), 60-70 (Average) 50-60 (Below Average), 0-50 (Not Passing).

Hint: use switch statement.

Aim: Understand the use of switch statement.

### Sample Answer:

```
// GradeReport.java
class GradeReport
{
    public static void main(String[] args)
    {
        int grade, category;
```

```

grade = Integer.parseInt(args[0]);

category = grade/10;

switch (category)
{
    case 10:
        System.out.println ("Perfect Score");
        break;
    case 9:
        System.out.println ("Excellent");
        break;
    case 8:
        System.out.println ("Good");
        break;
    case 7:
        System.out.println ("Above Average");
        break;
    case 6:
        System.out.println ("Average");
        break;
    case 5:
        System.out.println ("Below Average");
        break;
    default:
        System.out.println ("Not Passing");
        break;
}
}
}

```

- (d) Write a program to print all odd numbers between 1 and 20. Note : Use while loop.  
 Aim: Understand the use of while loop.

**Sample Answer:**

```

//OddNumbers.java

class OddNumbers
{
    public static void main(String[] args)
    {
        final int MAX_NUMBER = 20;
        int i = 1;

        while (i <= MAX_NUMBER)
        {
            if (i%2 > 0)
                System.out.println(i);
            i++;
        }
    }
}

```