University of Cape Town
Department of Computer Science

Computer Science CSC115F

June 2003 Final Exam

- Answer all questions.
- All questions that refer to elements of programming make reference to the Java programming language as studied in class.

Marks: 100
Time: 3 hours

- Approximate marks per question are shown in brackets
- The use of calculators is permitted

	Surname	Initials
NAME:		

STUDENT NO:

COURSE CODE: CSC

This paper consists of 15 questions and 21 pages (including this cover page).

Mark Allocation

Quest	Marks	Internal	External	Quest	Marks	Internal	External
1	[5]			9	[11]		
2	[5]			10	[7]		
3	[5]			11	[8]		
4	[5]			12	[5]		
5	[5]			13	[5]		
6	[15]			14	[5]		
7	[10]			15	[2]		
8	[7]						
	Total				**Total**		
					Grand Total		
					Final Mark		

Internal Examiner:	**External Examiner:**

Section 1. Introduction and Turtle Graphics

Question 1. **[5 marks]**

Explain the following computing concepts in your own words, give examples where appropriate.

a) Machine language

[1]

b) Parameter

[1]

c) Java Byte Code

[1]

d) A compiler

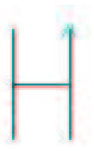
[1]

e) A boolean variable

[1]

Question 2. **[5 marks]**

Using the Turtle class write a class called “DrawShape” that draws the following shape. The final position of the turtle is not important.



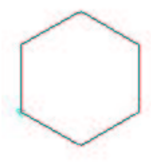
The Turtle instruction set:

- Move( ) // Moves the Turtle, if the pen is down leave a line.
- TurnRight( ) // Turns the turtle to the right. Max turn is 180 degrees.
- PenUp( ) // Picks up the Turtle pen.
- PenDown( ) // Puts down the Turtle pen.
- IsPenDown( ) // Returns true or false depending on the state of the Turtle.

[5]

Question 3. **[5 marks]**

Using the Turtle class write a method for a SmartTurtle class called "drawPolygon(sides, size)" that accepts 2 parameters, *sides* and *size*. The method should draw a regular polygon with the given number of *sides*, and all sides are of length *size*. Your answer should include the definition of the SmartTurtle class.
For example, drawPolygon(6, 200) produces:

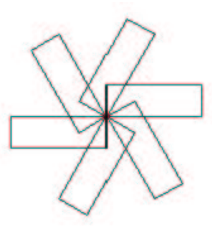


Use the Turtle instruction set from the previous question.

[5]

Question 4. **[5 marks]**

Write a java class called Shape to draw the following shape.



The size of each rectangle is 100 by 300 turtle steps. Use the information about the turtle given in the previous question. You should make your program as efficient as possible.

[5]

Question 5. [5 marks]

A programmer is writing a class using the Turtle class to draw a square (Note: the pen starts down). This is the first attempt and the compiler detects 5 errors.
Underline the errors, and show the correction in the space provided.

```
import turtlegraphics.*;

new class DrawSquare
{
  public static void main(  args)
  threw TurtleException
  {
    Turtle myTurtle = new Turtle();

    myTurtle.move(500);
    myTurtle.turnRight(90),
    myTurtle.move(500);
    myTurtle.turnRight(90);
    myTurtle.move(500);
    myTurtle.turnRight(90);
    myTurtle.move(500);
    myTurtle.turnRight(90);
  )
}
```

1.

2.

3.

4.

5.

[5]

Section 2. Java Basics

Question 6. **[15 marks]**

a) List the 3 syntax errors in the following code fragment (line numbers are added so you can refer to specific lines):

```
line1: public float func ( integer a, float b )
line2: {
line3:    float result = (1.0f+((a*2)+(b*3))
line4:    return result;
line5: }
```

[3]

b) List one difference between constants and literals.

[2]

c) Why does the expression `"day"<31` result in an error?

[1]

d) Write the method `calcRoot` to calculate a root of a quadratic equation (i.e., a value of x for which $ax^2 + bx + c = 0$). Your method must assume that a, b and c are `double` instance variables. `calcRoot` must take no parameters and return a floating point result. In addition, if $b^2 - 4ac$ is negative, the returned value must be 0 and an error message must be written to the console (screen).

Remember that the roots are given by: $x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$ and that the `Math.pow (x, y)` method calculates x^y.

[4]

e) Write a method `adjustedAverage` to calculate the average of the two highest marks in a set of three marks. For example, the adjusted average of 12, 24 and 36 is 30 because the smallest value is ignored. Your method must take 3 integer parameters and return an integer.

[5]

Section 3. Number Systems

Question 7. **[10 marks]**

a) In boolean addition, explain what an overflow is and illustrate with an example.

[3]

b) Convert 23.125_{10} into its hexadecimal representation. Show full calculations and clearly indicate your final answer.

[3]

c) Write an algorithm to add together 2 whole binary numbers in 1's complement, where the numbers have differing numbers of bits. Assume the numbers are already in 1's complement and leave the result in 1's complement form.

[4]

Section 4. Error Checking and Numerical Accuracy

Question 8. **[7 marks]**

The following program should print out the series $\frac{1}{10}, \frac{1}{9}, \frac{1}{8}, ..., \frac{1}{1}$ in decimal representation. However, the code contains some errors.

```
class Fractions
{
    public static void main(String[] args)
    {
      div = 10;
      while (div >= 0)
        {
         System.out.printline(1/div);
         div--;
        }
    }
}
```

a) What types of error(s) occur when the program is compiled and executed?

[2]

b) Identify each error and explain how to fix it.

[3]

c) What are *round-off* errors in floating-point numbers? Give an example of when one could occur.

[2]

Section 5. Using Java System Classes

Question 9. [11 marks]

Examine the program below:

```
import java.io.*;
import java.util.StringTokenizer;
import java.text.NumberFormat;
public class fileTest5
{
  public static void main(String[] args)
      throws java.io.IOException, java.text.ParseException
  {
    BufferedReader inStream
      = new BufferedReader(new InputStreamReader(System.in));
    NumberFormat aNumberFormatter = NumberFormat.getInstance();

    File inFile = new File("myFile.dat");
    if(inFile.exists() && inFile.canRead())
    {
       BufferedReader fileInStream
         =  new BufferedReader(new FileReader(inFile));
       String line = fileInStream.readLine();
       StringTokenizer st
           = new StringTokenizer(line);

       //start of code to replace
       int sum=0;
       while (st.hasMoreTokens())
       {
         String num = st.nextToken();
         System.out.print(num + "+");
         int numInt = aNumberFormatter.parse(num).intValue();
         sum = sum + numInt;
       }
       System.out.print("=" + sum);
       System.out.flush();
       // end of code to replace
       fileInStream.close();
    }
    else System.out.println("could not open input file");
  }
}
```

a) What does the NumberFormat class do?

[1]

b) If the input file *"myFile.dat"* contains the text:

```
40 50 60 50 100
60 70 50 40 20 20
```

write down the exact output produced by the program.

[2]

c) If the input file *"myFile.dat"* contains the text:

```
Mary had a little lamb
```

describe what the program does.

[1]

d) The *countTokens()* method of the *StringTokenizer* class returns the number of tokens remaining from the current token to the end of the string. Using the *countTokens()* method, replace the *while* loop in the program above with a *for* loop. Just write the replacement code.

[3]

e) Using the *countTokens()* method, write a program that outputs the number of words in each line in a file as well as the total number of words in the whole file. e.g. For the *"myFile.dat"* example above, the output would be:

```
No. words in line 1: 5
No. words in line 2: 6
Total: 11
```

You are given the program skeleton below, just write the missing code in the space provided on the following page.

```
import java.util.StringTokenizer;

public class fileTest2
{
    public static void main(String[] args)
      throws java.io.IOException
    {
      BufferedReader inStream
         = new BufferedReader(new InputStreamReader(System.in));

      System.out.print("Enter input file name: ");
      System.out.flush();
      String inputFileName = inStream.readLine();

      File inFile = new File(inputFileName);

      if(inFile.exists() && inFile.canRead())
        {
           BufferedReader fileInStream
              =  new BufferedReader(new FileReader(inFile));

           // YOUR CODE GOES HERE

           fileInStream.close();
        }
       else System.out.println("could not open input file");
    }
}
```

[4]

```
    }
```

Section 6. Writing Your Own Classes

Question 10. **[7 marks]**

A car hire company, “Best Cars”, wants a program to keep track of its vehicles. Examine the following skeleton definition for a class Car.

```
public class Car
{
    public Car()
    {
       this("",0,true);
    }

    public Car(String make, int odometer, boolean inGarage)
    {
        setData(make, mileage, inGarage);
        NoCars++;
    }

    public void setData(String make, int odometer, boolean inGarage)
    {
        //code missing
    }

    public void AddDistance(int distance)
    {
       odometer += distance;
    }

    public boolean CheckOut()
    {
      if (!inGarage) return false;
      else {
             inGarage = false;
             return true;
           }
    }

    public boolean CheckIn(int distance_traveled)
    {
      if (inGarage) return false;
      else {
             AddDistance(distance_traveled);
             inGarage = true;
             return true;
           }
    }
```

```
   private String make;
   private int odometer;
   private boolean inGarage;
   private static int NoCars=0;
}
```

a) What is the difference between **actual** parameters and **formal** parameters? Give examples of each using the class *Car*.

[2]

b) The body of the *setData* method is not defined. Write the complete code for this method:

[2]

c) Assume that all the methods in the class have been completely defined. Now show how you would create a brand new Mazda car for your garage, check it out for a rental and then check it in with 150 kilometers traveled.

[2]

d) Look carefully at the following method definition for the class Car and explain what is wrong with it.

```
public static void CheckAllIn()
{
   inGarage = true;
}
```

[1]

Section 7. Arrays and Software Engineering

Question 11. **[8 marks]**

Complete the Java fragment given below that:

a) reads the integer values that are guaranteed to be from 0-20 inclusive. Any number of integers can be read in with the sequence terminated by a -1 (negative 1).

b) keeps track of how many of each integer are read in (i.e. 3 zero's, 2 one's, 4-two's, etc).

c) outputs "Exact Sequence" if each integer 0-20 occurs exactly once.

```
public static void main (String[] args)
{
    boolean sequence=true;
```

[8]

```
}
```

Question 12. **[5 marks]**

Given the following Java fragment write a method that copies all the elements of an array "one" to an array "two". This method also rearranges the columns of "two" so that they are in reverse order (i.e. contents of column 0 of "one" are put in column 3 of "two", column 1 is put in column 2 and so on).
Show how you would call the method from the main program.
Example:

```
    one        two

 1 2 3 4    4 3 2 1
 5 6 2 1    1 2 6 5
 8 7 9 4    4 9 7 8
 5 4 1 6    6 1 4 5
```

```
public static void main (String[] args)
{
    final static int max = 4;
    int [][] one = {{1, 2, 3, 4}, {5, 6, 2, 1} ,
                    {8, 7, 9, 4} , { 5, 4, 1, 6}};
    int [][] two = new int[max][max];
```

[5]

```
}
```

Question 13. **[5 marks]**

Below you are given the class "Car". Use the concepts of inheritance and polymorphism to:

a) define a new class called "EstateCar" that has an extra variable called roofRackSize of type integer;

b) write a constructor for the new class; and

c) write the "ToString()" method for the new class.

```
public class Car
{
    private String name;
    private int bootSize;

    public Car (String n, int bs)
    {
        name = n;
        bootSize = bs;
    }

    public String ToString()
    {
        return "Name: " + name + " BootSize: "+bootSize;
    }
}
```

[5]

Question 14. **[5 marks]**

Consider a computer system for a bookseller. This bookseller keeps a stock of many books, magazines and comics. They are ordered from several different publishers and put into the stock. Each book, magazine and comic has only one publisher. The books, magazines and comics are sold.
Draw a UML class diagram giving:

a) classes invloved;

b) relationships between the classes; and

c) methods on the classes.

[5]

Question 15. **[2 marks]**

An operating system usually consists of 7 different layers, each of which performs a specific function. Name 4 of these layers.

[2]