University of Cape Town
Department of Computer Science

Computer Science CSC116S

Test 3 - 6 October 2004

- This test covers Number Systems, the Graphics class and Computer Architecture.
- Answer all questions.
- Good luck !

Marks: 35

Time: 40 minutes

- 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 6 questions and 7 pages (including this cover page).

Mark Allocation							
Quest	Marks	Internal	External	Quest	Marks	Internal	External
1	[3]			4	[2]		
2	[5]			5	[6]		
3	[7]			6	[12]		
	Total				**Total**		
					Grand Total		
					Final Mark		

Internal Examiner:	**External Examiner:**

Question 1. [3 marks]

Consider the Java statements below which write the strings “smile” and “nice” on a frame or canvas, and answer the questions that follow with respect to this code:

```
g.drawString("smile", 80, 240 );
g.drawstring("nice", 100, 200);
```

a) Will the string “nice” appear *above* or *below* the string “smile!” in the output ?
above [1]

b) Name any one other drawing method that object g has.

setFont(), setColor(),drawLine(), drawRect(), fill/drawPolygon(), fill/drawArc(), fill/drawOval() [1]

c) Which part of the letter “n” in “nice” will be located precisely at point 100, 200 and why do you think they arranged it that way? (i.e. give a reason for your answer)

bottom left - so that when you use drawLine at that point it underlines the whole word starting from the begining of ”n” [1]

Question 2. [5 marks]

Answer the questions that follow with respect to the simple Java code shown below:

```
import java.awt.*;
import java.event.*;
class TestExample extends Frame implements MouseListener
{
    public TestExample( )
    {
        super ("Test Example");
        addMouseListener ( this );
        setSize ( 400, 400 );
        show( );
    }

    public void mouseClicked ( MouseEvent event )
    { System.out.println ("mouse clicked"); }

    public void mousePressed ( MouseEvent event )
    { System.out.println ("mouse pressed" ); }

    public void mouseResleased ( MouseEvent event )
    { System.out.println ("mouse released" ); }

    public void mouseExited ( MouseEvent event ){ }
    public void mouseEntered ( MouseEvent event ){ }
```

```
    public static void main ( String[ ] args)
    {
        new TestExample( );
    }
} // end of TestExample class
```

a) If the user clicks the mouse once inside the frame when this program is run, what will the program output?
mouse pressed
mouse released
mouse clicked

[1]

b) What is the difference between “extends” and “implements” in line 3?

Keyword extends used to subclass
Keyword implements used to provide the code for an interface

[2]

c) Provide a statement for the mouseEntered method so that it prints out a message to indicate at which point (x and y co-ordinates) the mouse entered the frame:

System.out.println (”Mouse entered at ” + event.getPoint()); [1]

d) Can we use a drawOval statement in the body of mouseEntered to draw a small circle on the frame at the point where the mouse entered (Yes / No) ? Give a clear and complete reason for your answer.

No, because half of the circle will be out of the frame since the top-most point of the circle is at the edge of the frame [1]

Question 3. [7 marks]

a) Convert the decimal number 52 to binary:

```
52/2 = 26 r 0
26/2 = 13 r 0
13/2 = 6  r 1
6/2  = 3  r 0
3/2  = 1  r 1
1/2  = 0  r 1
 Ans = 110100
```

[2]

b) **DO EITHER i OR ii**

i) Give the two's complement binary representation of the decimal value -52
OR

ii) Convert the octal number 52 to decimal

i:

```
-52 = 2sComplement(110100)
    = 001100
```

ii:

```
52base8 = 5*8^1 + 2*8^0
        = 40 + 2
        = 42 base 10
```

[1]

c) Show below how the decimal number -52 would be represented in IEEE single precision format, i.e. where there is one sign bit, then an 8-bit biased exponent, and finally a 23-bit significand:

```
       S  = 1 (negative sign)
       52 = 110100
Normalise : 110100 = 1.101*2^5
Exponent  : 5 + 127 (bias) = 132 = 10000100
      Ans = 1 10000100 10100...0000 23 bits
```

[4]

Question 4. [2 marks]

Explain what the following registers are used for:

a) Program counter
Holds address of next instruction to execute

[1]

b) Instruction counter
Holds the current instruction being obeyed

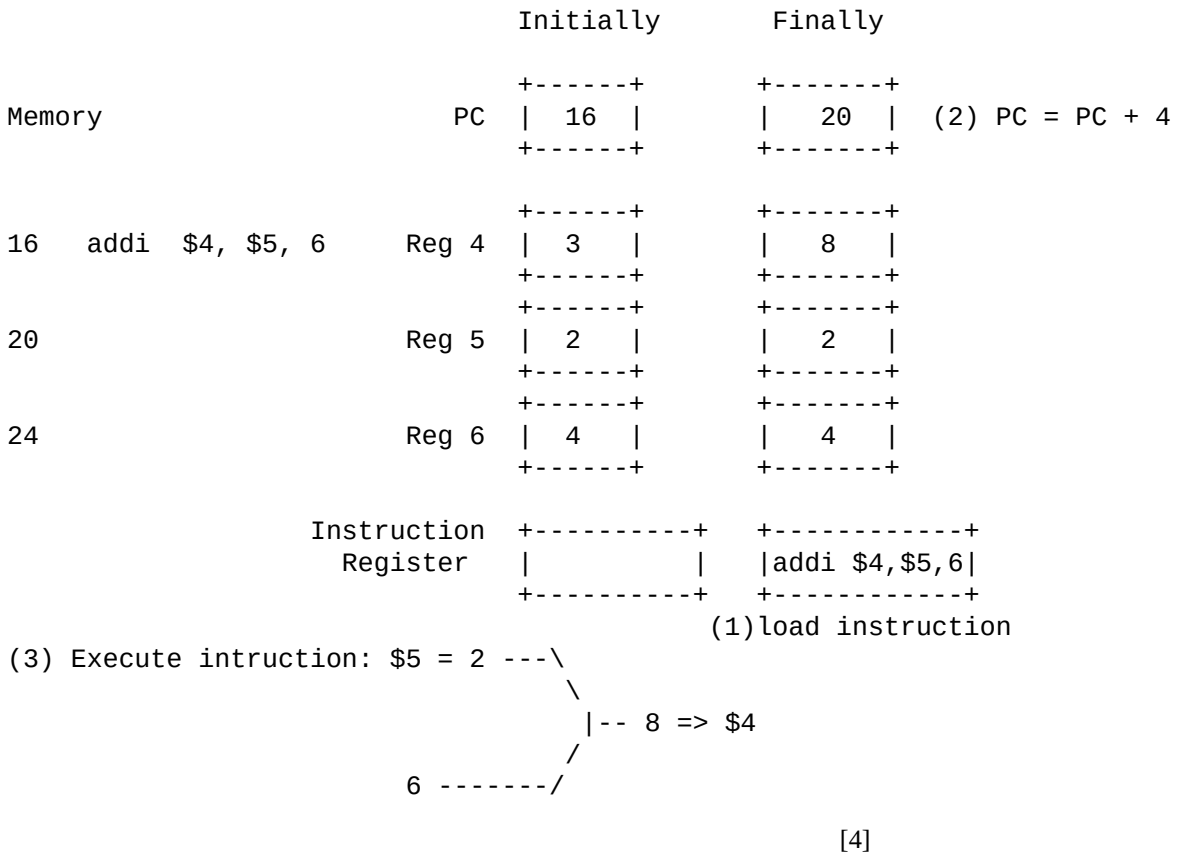
[1]

Question 5. [6 marks]

a) Explain the steps that the control unit causes the computer to perform.
1. Load instruction
2. PC = PC + 4
3. Execute instruction

[2]

b) Illustrate your answer for the execution of the next instruction given the state of the computer below. Show how the relevant registers of the computer change at each step.



[4]

Question 6. [12 marks]

Write a MIPS assembler program to do the same as the following java program:

```
import Keyboard;
public class FindMax
 {
    public static void main(String[] args)
    throws java.io.IOException
    {
        int x, big = 0;

        for (int i=0; i<10; i++)
        {
            x = Keyboard.readInt();
            if (x > big)
            { big = x; }
        }

        System.out.println("Big value = " + big );
    }
}
```

Comment all MIPS statements.

Given:

```
            .data
      x:    .word 0
    big:    .word 0
      i:    .word 0
            .text
            .globl main
```

(continued on the next page)

```
   main:

         lw      $s0, big        # $s0 = big = 0
         addi    $t0, $0, 0      # initialise loop variables
         addi    $t1, $0, 10

loop:    bge     $t0, $t1, exit  # loop for i < 10,  OR  blt $t1,$t0, exit
         addi    $t0, $t0, 1     # increment i
         addi    $v0, $0, 5      # prepare to read an int
         syscall                 # answer stored in $v0

         ble     $v0, $s0, loop  # check if x > big  (many variations of it)
         add     $s0, $s0, $v0   # big = x
         sw      $s0, big        # store to memory
         j       loop

exit:    addi    $v0, $0, 1      # prepare to print and int
         add     $a0, $0, $s0    # put big in $a0
         syscall

         jr      $ra             # exit program
```

[12]