

**CPSC 321:501-503 - Computer Architecture**  
**Texas A&M University**  
**Department of Computer Science**

**Fall 2006**

**Project 1 (100 pts) - Complete in groups of 2**

---

**Release date:** 2 October 2006

**Due date:** 16 October 2006

---

## 1 Objective

The objective of the project is to develop an interactive program to render various shapes (lines, circles, rectangles and polygons) onto the console, based on inputs from the user.

---

## 2 Specification

You are required to develop a *menu-driven interactive* MIPS program that will allow the user to instantiate a fixed size drawing board and render various shapes - lines, circles, rectangles and polygons - onto it. This drawing board will then be displayed onto the console after every draw operation.

The size of the drawing board, shapes to be drawn and their specifications will be taken in as input from the console.

**Note:** The origin is to be set in the bottom left corner of the drawing board.

The following functions are to be implemented for the completion of the project:

1. `void initDrawingBoard(int height, int width)`

This function should allocate space for the drawing board (`height x width` bytes) and initialize the values on the drawing board.

2. `void drawLine(int x1, int y1, int x2, int y2, char a)`

This function should draw a line from  $(x_1, y_1)$  to  $(x_2, y_2)$ . The line should be drawn with character `a`.

3. `void drawCircle(int x, int y, int r, int fill, char a)`

This function should draw a circle with center at  $(x, y)$  and a radius of `r`. If `fill==1`, fill circle with character `a`, else if `fill==0`, do not fill circle.

4. `void drawRect(int x1,int y1,int x2,int y2,int fill,char a)`

This function should draw a rectangle with  $(x1,y1)$  and  $(x2,y2)$  as the opposite corners. If `fill==1`, fill rectangle with character `a`, else if `fill==0`, do not fill rectangle.

5. `void drawShape(int x,int y,char *lineSeq,int fill,char a)`

This function should draw a shape starting at  $(x,y)$ , and using the line sequence `lineSeq` specified. `fill` will be used as stated above. The line sequence will be specified as a  $\langle \text{direction} \diamond \text{distance} \rangle$  pair. The possible directions are north(n), south(s), east(e), west(w), north-east(ne), north-west(nw), south-east(se) and south-west (sw).

### For example:

Consider the line sequence string "n10e15s10w15" would draw a rectangle 10 x 15 units.

```
*****
*           *           *****
*           *           *****
*           *           *****
*           *           *****
*           *           *****
*           *           *****
*           *           *****
*           *           *****
*           *           *****
*****
fill = 0           fill = 1
```

The line sequence string "ne3se3w5" would draw a triangle.

```
*           *
* *         ***
*****     *****
fill=0     fill=1
```

Test your code with more complex shapes.

**Note:** In this case, `fill` will only make sense when the shape is closed, i.e. the line sequence finishes the shape back at the *starting point*.

6. `void drawString(int x,int y,char *string)`

This function should draw string at location  $(x,y)$ .

7. `void printDrawingBoard()`

This function should display the contents of the drawing board onto the console.

Your menu should have an *exit* option to indicate that the user does not wish to draw anymore shapes.

---

## 3 Deliverables

### **Design Notebook (30pts)**

Please keep a design notebook that documents your design process and all the steps in the program development; it should document the progress that you made while writing your software (keep track of date and time). Write the design notebook while you are developing the software, do not write it after the fact.

Your notebook should keep a log of all the errors that you make; it should be handwritten with a pen that is not erasable; printouts documenting your progress should be glued in. The grading of your design notebook will take into account the clarity, regularity, legibility and organization of your annotations.

### **Assembly Program (40pts)**

Turn in your assembly program via the *turnin* command. Ensure that your program is clearly commented. **Note: Be aware of the *register conventions*. It will make programming much simpler.**

### **Project Demonstration (30pts)**

Project demonstrations should be completed within a week after submission. Failure to demonstrate will result in a Fail grade on the project.

---

## 4 Dishonesty

Make sure that you complete the assignment by yourself. Do not copy the code from others, nor provide others with your code. Refrain from copying and modifying the code from other sources.

---