ASSIGNMENT 1

MELVIN RIJOHN T

10/12/2024

1. Class Inheritance

```
class Shape;
    string name;

    function new(string name);
        this.name = name;
    endfunction

    function void print();
        $display("Shape: %s", name);
    endfunction
endclass

class Circle extends Shape;
    real radius;

    function new(real radius);
        super.new("Circle");
        this.radius = radius;
    endfunction

    function real calc_area();
        return 3.1416 * radius * radius;
    endfunction
endclass

class Rectangle extends Shape;
    real length, width;

    function new(real length, real width);
        super.new("Rectangle");
        this.length = length;
        this.width = width;
    endfunction

    function real calc_area();
        return length * width;
    endfunction
endclass

module test;
    Circle c;
    Rectangle r;
    initial begin
        c = new(5.89);
        c.print();
        $display("Area of %s: %0.2f", c.name, c.calc_area());
```

```
        r = new(4.25, 7.16);
        r.print();
        $display("Area of %s: %0.2f", r.name, r.calc_area());
    end
endmodule


// OUTPUT
// # KERNEL: Shape: Circle
// # KERNEL: Area of Circle: 108.99
// # KERNEL: Shape: Rectangle
// # KERNEL: Area of Rectangle: 30.43
```

2. Semaphore

```
module process_ipc_semaphore ();
    semaphore sem1; // = new(2);
    semaphore sem2; // = new(4);

    task process(string name);
        $display("[%0t]: Process %0s trying to access semaphore key.", $time, name);
        sem1.get(1);
        $display("[%0t]: Process %0s accessed semaphore key.", $time, name);
        #10;
        sem1.put(1);
        $display("[%0t]: Process %0s released semaphore key.", $time, name);
    endtask
    initial begin
        sem1 = new(2);
        sem2 = new(4);
    end
    initial begin
        fork
            process("P1");
            process("P2");
            process("P3");
        join
    end

    initial begin
        $display("[%0t]: Process P4 trying to acquire 3 keys", $time);
        sem2.get(3);
        $display("[%0t]: Process P4 acquired 3 keys", $time);
        #10;
        sem2.put(1);
        $display("[%0t]: Process P4 released 1 key", $time);
        #5;
        sem2.get(2);
        $display("[%0t]: Process P4 released 2 key", $time);
```

```
    end

    initial begin
        $display("[%0t]: Process P5 trying to acquire 2 keys", $time);
        sem2.get(2);
        $display("[%0t]: Process P5 acquired 2 keys", $time);
        #10;
        sem2.put(2);
        $display("[%0t]: Process P5 released 1 key", $time);
    end

endmodule

// OUTPUT
//# KERNEL: [0]: Process P1 trying to access semaphore key.
// # KERNEL: [0]: Process P1 accessed semaphore key.
// # KERNEL: [0]: Process P2 trying to access semaphore key.
// # KERNEL: [0]: Process P2 accessed semaphore key.
// # KERNEL: [0]: Process P3 trying to access semaphore key.
// # KERNEL: [0]: Process P4 trying to acquire 3 keys
// # KERNEL: [0]: Process P4 acquired 3 keys
// # KERNEL: [0]: Process P5 trying to acquire 2 keys
// # KERNEL: [10]: Process P3 released semaphore key.
// # KERNEL: [10]: Process P3 released semaphore key.
// # KERNEL: [10]: Process P4 released 1 key
// # KERNEL: [10]: Process P3 accessed semaphore key.
// # KERNEL: [10]: Process P5 acquired 2 keys
// # KERNEL: [20]: Process P3 released semaphore key.
// # KERNEL: [20]: Process P5 released 1 key
// # KERNEL: [20]: Process P4 released 2 key
```