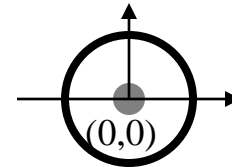
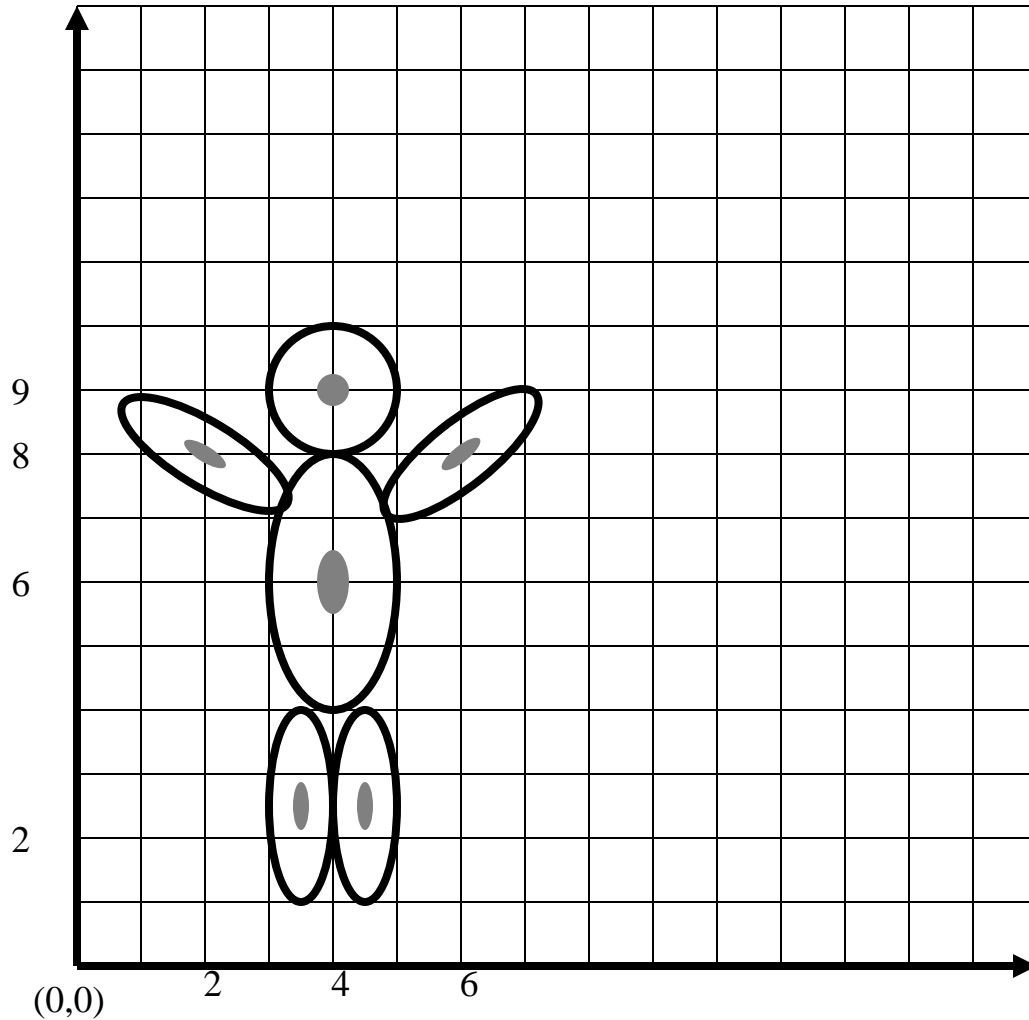
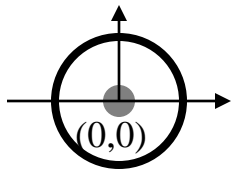


# Drawing with OpenGL



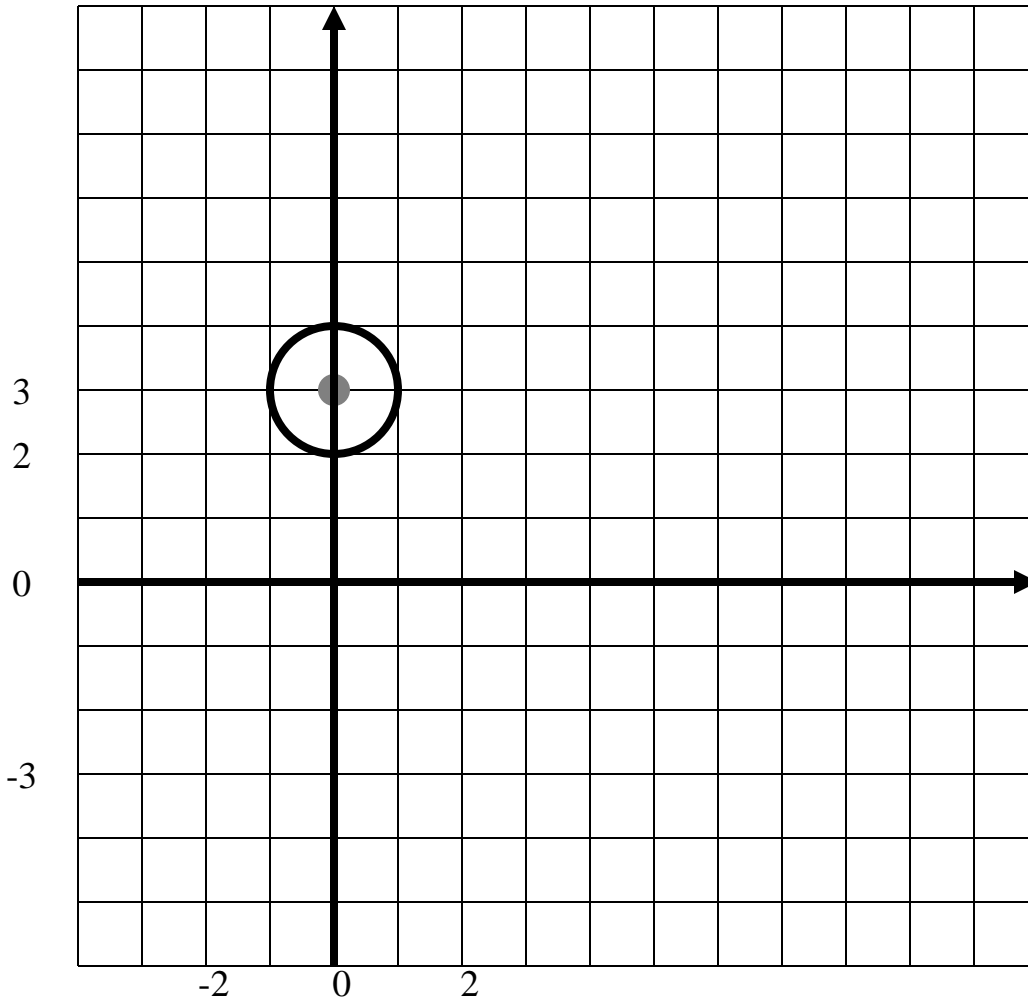
`DrawCircle();`



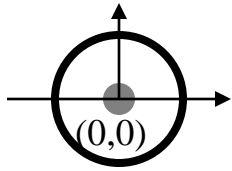
```
glMatrixMode(ModelView);  
glLoadIdentity();  
glPushMatrix();
```

```
//head
```

```
glPushMatrix();  
    glTranslate(0, 3, 0); DrawCircle();  
glPopMatrix();
```



```
glPopMatrix();
```



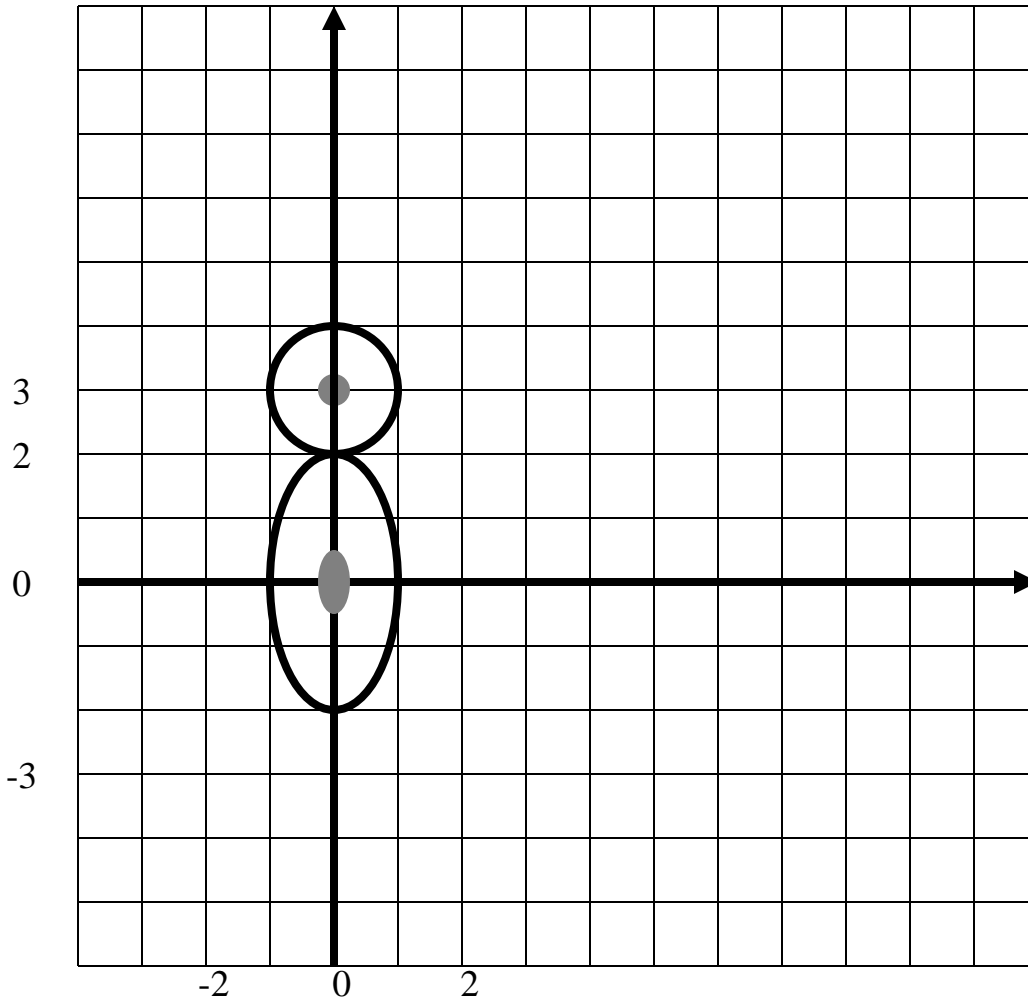
```
glMatrixMode(ModelView);  
glLoadIdentity();  
glPushMatrix();
```

```
//head
```

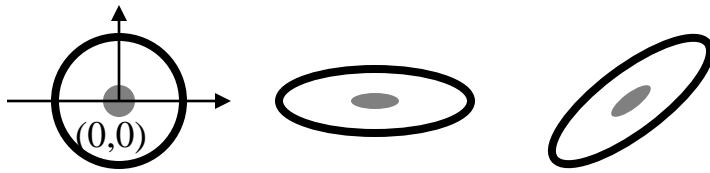
```
glPushMatrix();  
    glTranslate(0, 3, 0); DrawCircle();  
glPopMatrix();
```

```
//body
```

```
glPushMatrix();  
    glScale(1, 2, 1); DrawCircle();  
glPopMatrix();
```



```
glPopMatrix();
```



```
glMatrixMode(ModelView);
glLoadIdentity();
glPushMatrix();
```

**//head**

```
glPushMatrix();
  glTranslate(0, 3, 0); DrawCircle();
glPopMatrix();
```

**//body**

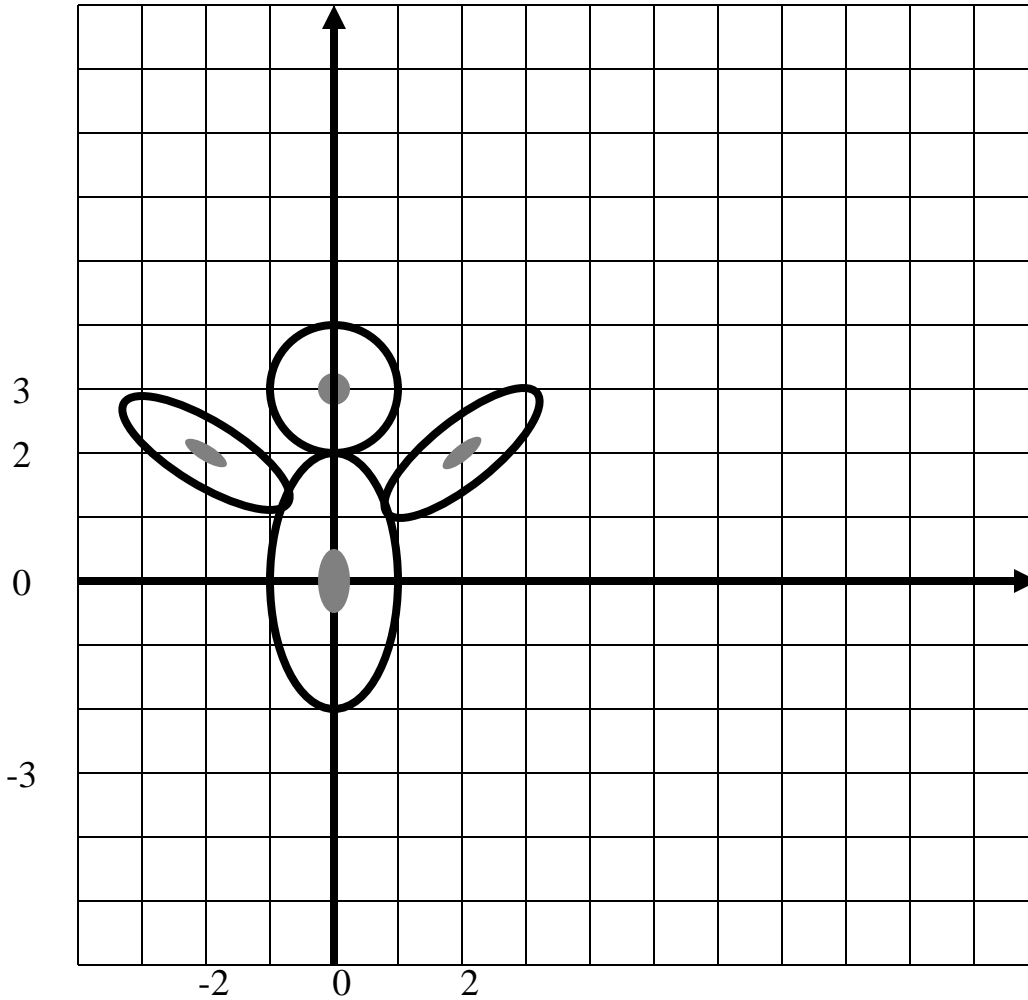
```
glPushMatrix();
  glScale(1, 2, 1); DrawCircle();
glPopMatrix();
```

**//left arm**

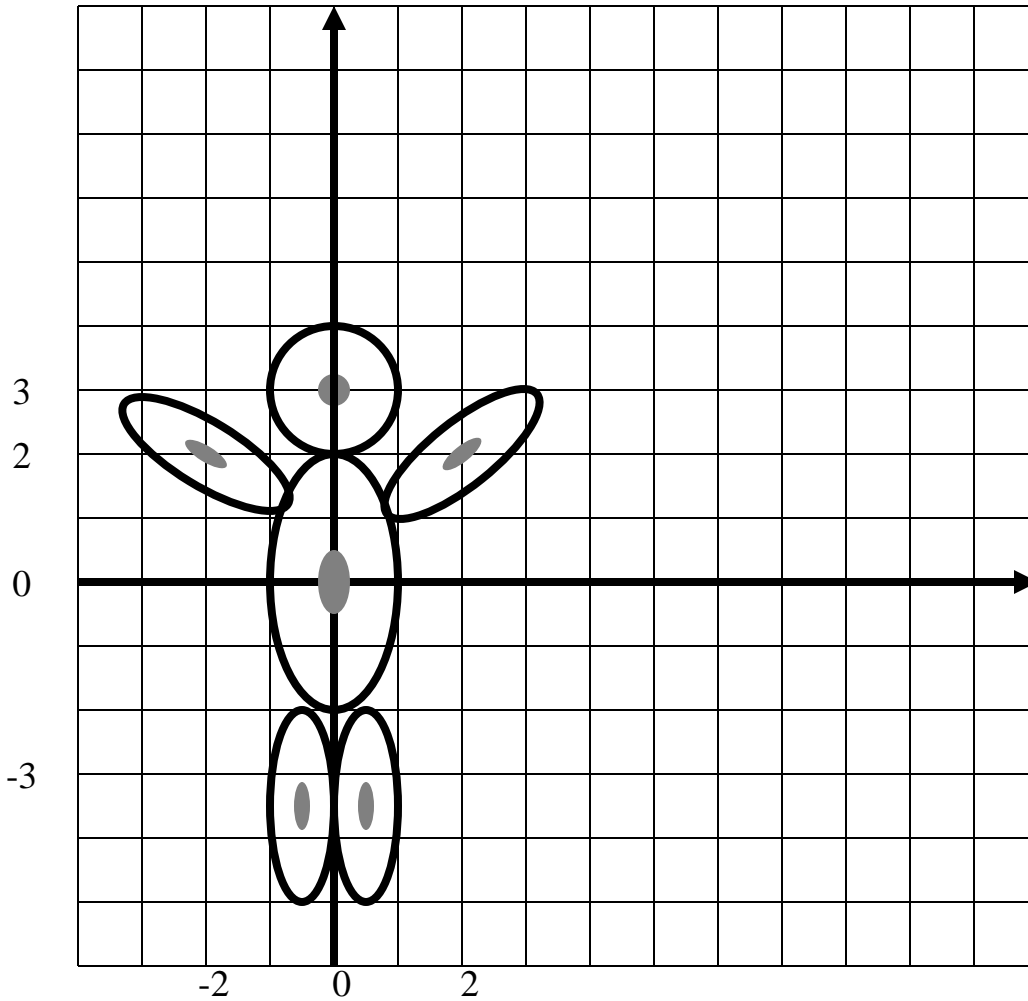
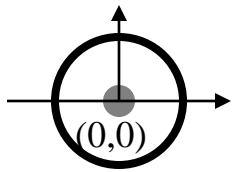
```
glPushMatrix();
  glTranslate(2, 2, 0);
  glRotate(45, 0, 0, 1);
  glScale(1.5, 0.5, 1); DrawCircle();
glPopMatrix();
```

**//right arm**

```
glPushMatrix();
  glTranslate(-2, 2, 0);
  glRotate(-45, 0, 0, 1);
  glScale(1.5, 0.5, 1); DrawCircle();
glPopMatrix();
```



```
glPopMatrix();
```



```
glMatrixMode(ModelView);  
glLoadIdentity();  
glPushMatrix();
```

```
//head
```

```
glPushMatrix();  
    glTranslate(0, 3, 0); DrawCircle();  
glPopMatrix();
```

```
//body
```

```
glPushMatrix();  
    glScale(1, 2, 1); DrawCircle();  
glPopMatrix();
```

```
//left arm
```

```
glPushMatrix();  
    glTranslate(2, 2, 0);  
    glRotate(45, 0, 0, 1);  
    glScale(1.5, 0.5, 1); DrawCircle();  
glPopMatrix();
```

```
//right arm
```

```
glPushMatrix();  
    glTranslate(-2, 2, 0);  
    glRotate(-45, 0, 0, 1);  
    glScale(1.5, 0.5, 1); DrawCircle();  
glPopMatrix();
```

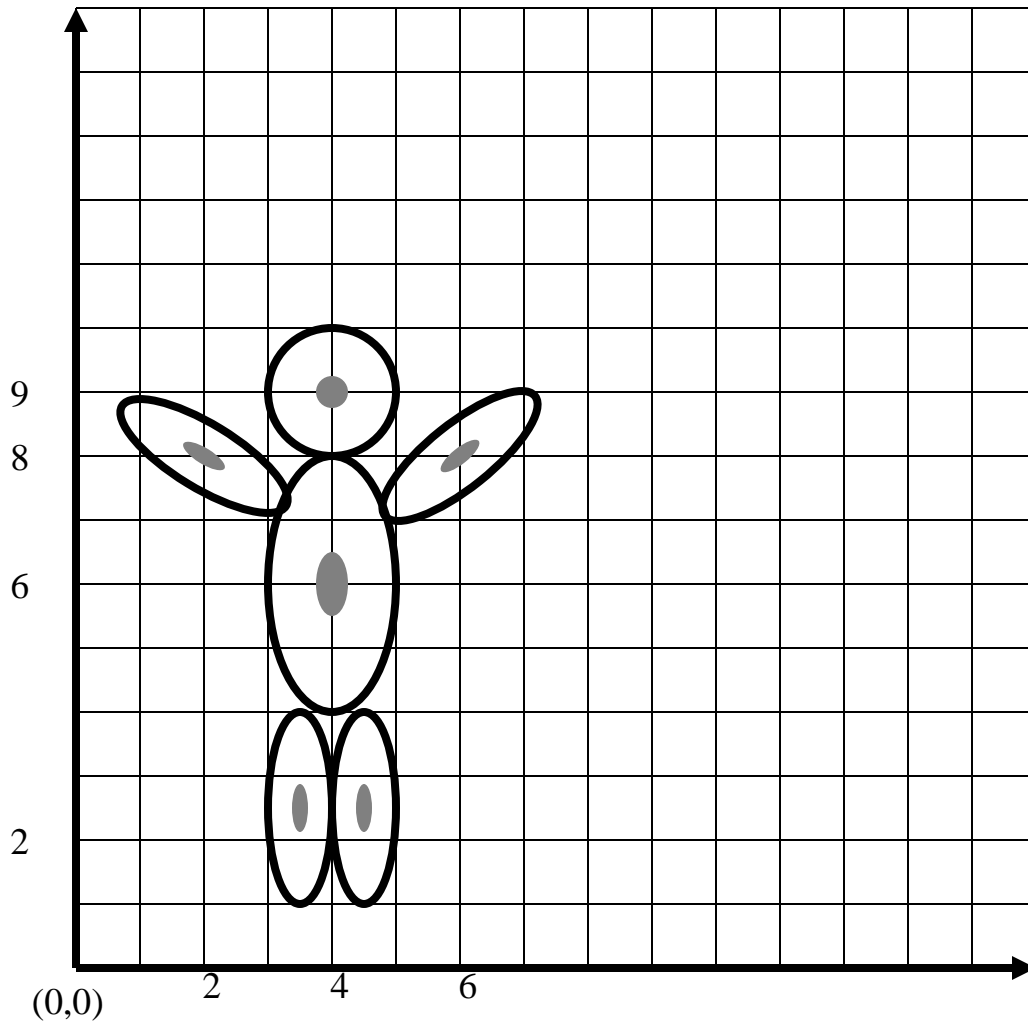
```
//left leg
```

```
glPushMatrix();  
    glTranslate(0.5, -3.5, 0);  
    glScale(0.5, 1.5, 1); DrawCircle();  
glPopMatrix();
```

```
//right leg
```

```
glPushMatrix();  
    glTranslate(-0.5, -3.5, 0);  
    glScale(0.5, 1.5, 1); DrawCircle();  
glPopMatrix();
```

```
glPopMatrix();
```



```
glMatrixMode(ModelView);
glLoadIdentity();
glPushMatrix();
```

```
glTranslate(4, 6, 0);
```

```
//head
```

```
glPushMatrix();
    glTranslate(0, 3, 0); DrawCircle();
glPopMatrix();
```

```
//body
```

```
glPushMatrix();
    glScale(1, 2, 1); DrawCircle();
glPopMatrix();
```

```
//left arm
```

```
glPushMatrix();
    glTranslate(2, 2, 0);
    glRotate(45, 0, 0, 1);
    glScale(1.5, 0.5, 1); DrawCircle();
glPopMatrix();
```

```
//right arm
```

```
glPushMatrix();
    glTranslate(-2, 2, 0);
    glRotate(-45, 0, 0, 1);
    glScale(1.5, 0.5, 1); DrawCircle();
glPopMatrix();
```

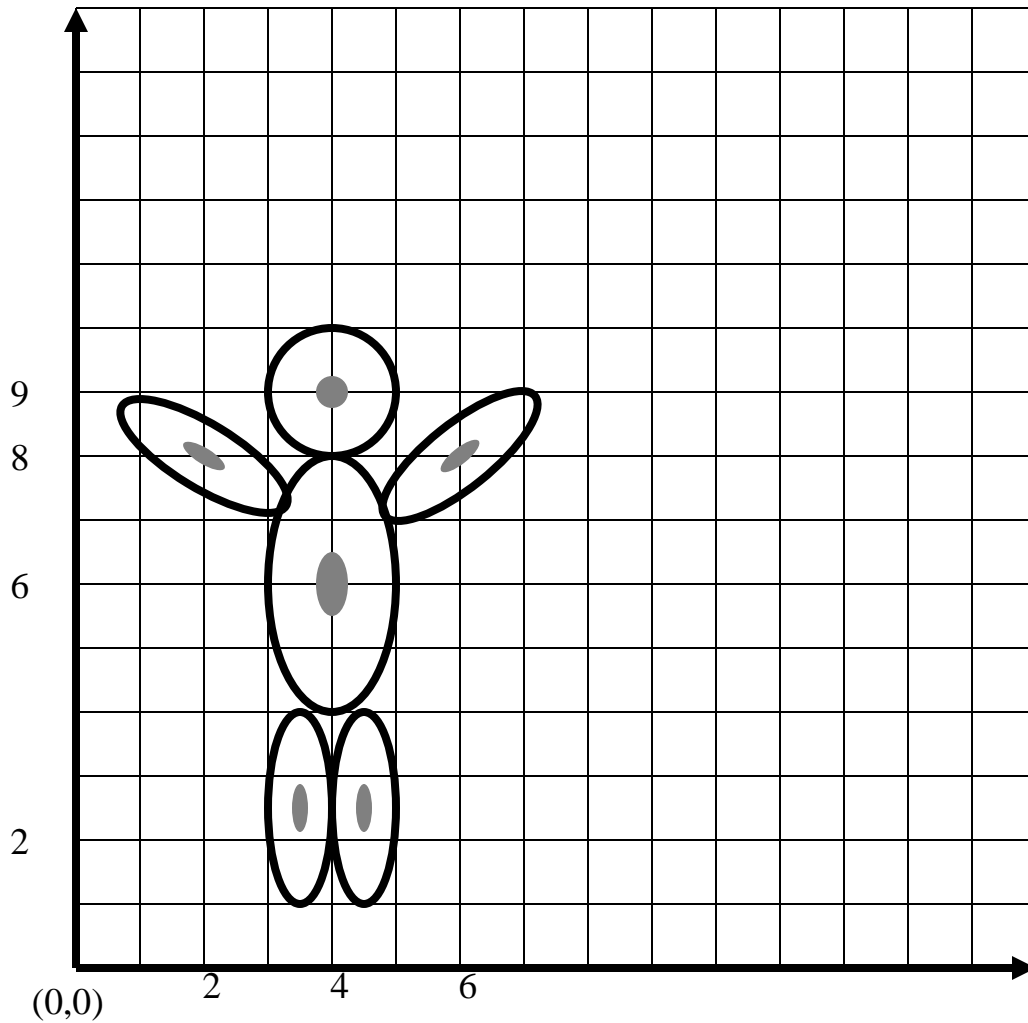
```
//left leg
```

```
glPushMatrix();
    glTranslate(0.5, -3.5, 0);
    glScale(0.5, 1.5, 1); DrawCircle();
glPopMatrix();
```

```
//right leg
```

```
glPushMatrix();
    glTranslate(-0.5, -3.5, 0);
    glScale(0.5, 1.5, 1); DrawCircle();
glPopMatrix();
```

```
glPopMatrix();
```



```

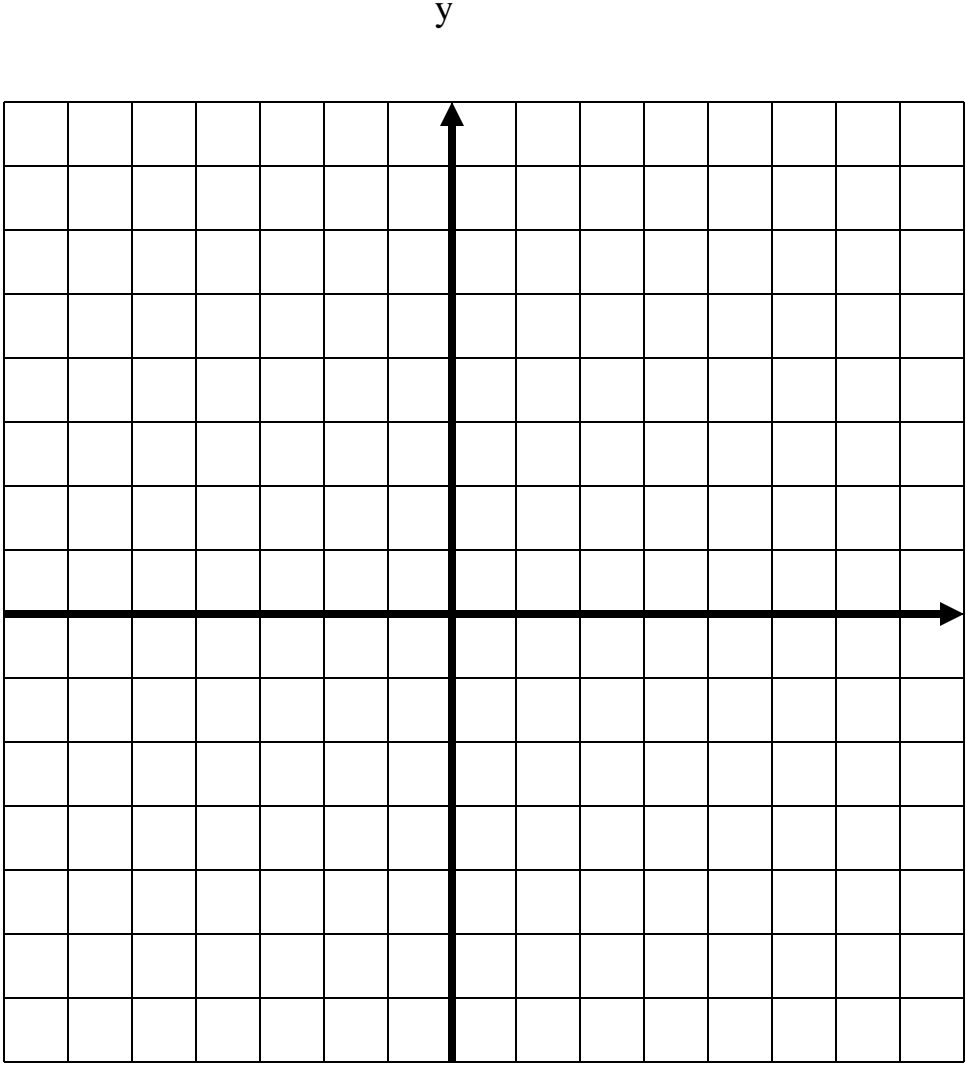
glMatrixMode(ModelView);
glLoadIdentity();
glPushMatrix();
    glTranslate(4, 6, 0);
    //head
    glPushMatrix();
        glTranslate(0, 3, 0); DrawCircle();
    glPopMatrix();
    //body
    glPushMatrix();
        glScale(1, 2, 1); DrawCircle();
    glPopMatrix();
    //left arm
    glPushMatrix();
        glTranslate(2, 2, 0);
        glRotate(45, 0, 0, 1);
        glScale(1.5, 0.5, 1); DrawCircle();
    glPopMatrix();
    //right arm
    glPushMatrix();
        glTranslate(-2, 2, 0);
        glRotate(-45, 0, 0, 1);
        glScale(1.5, 0.5, 1); DrawCircle();
    glPopMatrix();
    //left leg
    glPushMatrix();
        glTranslate(0.5, -3.5, 0);
        glScale(0.5, 1.5, 1); DrawCircle();
    glPopMatrix();
    //right leg
    glPushMatrix();
        glTranslate(-0.5, -3.5, 0);
        glScale(0.5, 1.5, 1); DrawCircle();
    glPopMatrix();
glPopMatrix();

```

# Exercise

- Use the locally `DrawCircle()` and `DrawSpline()` listed in the next slide and the handout to draw the final rendering





x

