

## Quiz #5

1. What are the advantages of using exceptions instead of return values to inform the calling code that an error has occurred?

**There are three basic advantages to using exceptions. First, checked exceptions can't be ignored. For those exceptions, the code won't even compile unless you put in a handler. Second, even for unchecked exceptions, the errors don't propagate. This implies that the code stops and notifies the user on the line where the problem happens if it isn't handled. In other schemes, the code can keep running for hundreds of lines before the problem actually causes it to crash. Third, exceptions can hold extra information about the problem and good exceptions generally tell you all the information that you would ask to help you debug a problem.**

2. Describe the steps that you would need to take to add a component to a GUI that will do custom drawing.

**If you want to add custom drawing to your GUI, you first have to create your own class that extends JPanel. Then go into that class and override the paintComponent method. That method will be called when the component should draw itself and it is passed a Graphics object. Anything you draw to that graphics object will appear on the screen when the panel is added into your GUI.**

Extra credit: On the back, write a function that will tell you if there is a non-self intersecting cycle of length >2 in a maze. That is to say that it returns a boolean which is true only if there is a path through the maze where you leave the current square and then return to it after taking more than 2 steps without crossing your path at any point except when you return to the starting square.

```
// This code assumes that the maze uses -1 for walls and 0 for open
// space. It should be called with a positive value in len.
boolean has2Cycle(int[][] maze,int x,int y,int len) {
    if(x<0 || x>maze.length || y<0 || y>maze[x].length) return false;
    if(maze[x][y]>0 && maze[x][y]<len-1) return true;
    if(maze[x][y]!=0) return false; // short cycle or wall.
    boolean ret=false;
    maze[x][y]=len;
    ret=has2Cycle(maze,x+1,y,len+1);
    if(!ret) ret=has2Cycle(maze,x-1,y,len+1);
    if(!ret) ret=has2Cycle(maze,x,y-1,len+1);
    if(!ret) ret=has2Cycle(maze,x,y+1,len+1);
    maze[x][y]=0;
    return ret;
}
```