Exam 2 Notes **Fall 2012**

1.

label	address	value
what[0]	1020	18
what[1]	1024	12
what[2]	1028	18
what[3]	1032	14
what[4]	1036	10
ptr	1040	~~1000~~ 1004
offset	1044	4

2.

Label	Address	Value
a.d	100	
a.str	108	
a.f	112	
a.c[0]	116	
a.c[1]	117	
a.c[2]	118	
a.c[3]	119	
a.c[4]	120	
t	121	~~100~~ 121
i	125	~~100~~ 22
j	129	~~121~~ 122
	133	

d = 22

3.

LABEL	ADDRESS	Value
st1	1000	1012
d	1004	
st2[0].i	1012	5
st2[0].d	1016	~~4.0~~ 6.6
st2[0].e	1024	~~1060~~ 1016
st2[0].j	1028	3
st2[1].i	1032	2
st2[1].d	1036	6.6
st2[1].e	1044	
st2[1].j	1048	
i	1052	20
st3.i	1056	

LABEL	ADDRESS	Value
st3.d	1060	4.4
st3.e	1068	
st3.j	1072	~~1032~~ 1028
g	1076	1060
e[0]	1080	1.1
e[1]	1088	
	1096	

4.

Explain why there is a memory leak.

5.

```
        struct s *mystruct = (struct s *)malloc(sizeof(struct s));
        mystruct->d = 1.0;
        mystruct->i = 2;
        mystruct->a = (char *) malloc((strlen("CLEMSON")+1) * sizeof(char));
        strcpy(mystruct->a, "CLEMSON");
```

6.

Must use pass-by-reference. Changes in red.

```
void division(int  numerator, int  denominator,
              int *dividend,  int *remainder    )
{
       if (denominator == 0) return;
       *dividend   =  numerator  /  denominator  ;
       *remainder  =  numerator  %  denominator  ;
}
int main(void)
{
       int  x, y ;
       int  div  ;
       int  rem  ;
       x  =  9;
       y  =  2;
       division (  x   ,  y   , &div   , &rem   );

       printf("%d/%d = %d with %d remainder\n", x , y , div , rem );
}
```

7.

```
double euclidian(struct Point p, struct Point q)
{
   return sqrt(  pow(p.x - q.x, 2)
              + pow(p.y - q.y, 2) );
}
```

8.

```
void calculate_perimeter( struct Triangle * T)
{
   T->perimeter =   euclidian( T->p1, T->p2)
                  + euclidian( T->p2, T->p3)
                  + euclidian( T->p3, T->p1);
}
```