CSE 20211 – Exam 1 – Fall 2012 NAME:

NETID:

Expressions. Evaluate each of these expressions and determine the type and the value. **(5 points)**

Variables	Expression	Type	Value
int a = 23, b = 5; double d = 10.0;	a % b +d		
double x = 10.0; short a=5, b =10;	a / b * 5.0 * x		
double a=100, b=1;	! (a > b)		
int a=2, b= 3;	++a + b++		
int a=0;	(a++) ? 'x' : 'z'		

Debugging. The following program is meant to play a simple guessing game with the user. Circle all of the errors in the program and explain the error in a few words. The errors may be compiling, linking, or logic errors. (5 points)

```
include <stdio.h>

void main ( int )
{
      secret = rand()%100;

      while( true ) {
            printf("Guess my number: \n");
            scanf("%c",n);

            if( n<secret ) {
                  printf("Too low!");
            } else if( n<secret ) {
                  printf("Too high!");
            } else {
                  printf("Correct!\n");
                  continue;
            }
}
```

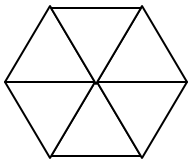
For each of the code fragments below, write what the program displays in the box to the right. Unless otherwise specified, assume that all variables have been declared as integers. (10 points)

```
int i,j;

for( i=0; i<5; i++ ) {
    for( j=0; j<5; j++) {
        if( (i+j) == 4 ){
            printf("X");
        } else if( (i-j) == 0 ) {
            printf("O");
        } else {
            printf(".");
        }
    }
    printf("\n");
}
```

```
int n = 31;

do {
    switch(n%10) {
        case 1:
        case 2:
        case 3:
           printf("up\n");
           n++;

        case 4:
           printf("jump\n");
           n *= 2;
           break;
        default:
           printf("down\n");
           n--;
           break;
   }
   printf("%d\n",n);

} while( n%5 != 0 );
```

(A good answer can fit in the available space, but use the back of the page if necessary.)

Using the gfx library, write a **function** that draws a polygon consisting of n triangles, centered at a location x,y, with a radius of r. For example, your function would draw this for n=6: (5 points)



Write a **function** that returns the following approximation for sin(x). (5 points)

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \ldots$$

*A number is **perfect** it its factors, including 1 (but not the number itself) sum to that number. For example, 6 is a perfect number because the factors of 6 are 2 and 3, and 6 = 1 + 2 + 3. 28 is a perfect number because 28 = 1 + 2 + 4 +7 + 14.*

Write a **program** that determines all the perfect numbers, starting with the smallest.
(Of course, it will run forever.) The output should be like this: (10 points)

```
6 is a perfect number
28 is a perfect number
496 is a perfect number
…
```