

You should be able to correctly answer all questions on this test prior to starting EEL3801. Answer these questions and check your answers with the answers provided on the web. If you missed a few questions but feel comfortable and understand why you missed them, then I recommend that you stay with the course but study the C++ book. If you missed a considerable part of the test you may consider taking EGN3210 first before taking EEL3801. If you missed the question dealing with f2 in #1 (the \*p got you) this is common and will be covered in class in more detail. If you missed #4 (files) this is also common. It will also be covered again.

#1 **(Functions 8%)** What is the output of the following program.

```

int    f1 (int a)           <-- gets 42
    {
    if (a == 37)           <-- false
        return 1;
    else if (a > 0)       <-- true
        return 2;       <-- returns 2
    else
        return 3;
    }

void   f2 (int a, int *p)
    {
    if (a < 42)           <-- false
        *p = 1;
    else if (a > 43)      <-- false
        *p = 2;
    else
        *p = 3;         <-- sets y in main to 3
    }

void   main( )
    {
    int    x, y, numb = 42;

    x = f1(numb);
    f2( numb, &y);
    printf("x = %d y = %d numb = %d \n", x, y, numb);
    }

```

output  
x = 2 y = 3 numb = 42

#2 **(Functions)** Write a function that implements the following:

$$f(x) = 5x^3 - 3x^2 + 2x + 7$$

Hint: Use  $x*x*x$  for  $x^3$  and  $x*x$  for  $x^2$

```
double f(double x)
{
    return 5*x*x*x - 3*x*x + 2*x + 7;
}
```

- #4 **(File)** Write a small program to ask the user for a file name, open the file for input and read every string until the end of file and print them to the screen. Use a while loop to read and print the lines and use the eof( ) function to detect the end of file.

```
void display()
{
    FILE *fp;
    char name[80],str[80];

    printf("enter a file name\n");
    scanf("%s",name);

    fp = fopen(name,"r");

    if (fp == NULL)
        return;

    while ( !eof(fp) )
    {
        fgets(str,80,fp);
        printf("%s",str);
    }

    fclose(fp);
}
```

#5 **(Structures)** Given the data structure below, create a variable called bill of type date and initialize the variable to 11/14/1968.

```
struct date
{
    int      month;
    int      day;
    int      year;
}
```

```
struct    date bill;
```

```
bill.month = 11;
bill.day = 14;
bill.year = 1968;
```

#6 **(Scope)** What is the output. Note: 0 is false and any value not 0 is true.

```
int    a = 1, b = 2, c = 3;
void main()
{
    int    b = 4;

    if ( 1 ) // 1 is true
    {
        int    c = 5;

        printf ("a = %d, b = %d, c = %d \n", a, b, c);
    }

    printf ("a = %d, b = %d, c = %d \n", a, b, c);
}
```

output

a = 1 b = 4 c = 5

a = 1 b = 4 c = 3

#7 (Arrays) What is the output of the following program.

```
void main( )
{
    int    A[5] = {10, 20, 30, 40, 50},
           B[5] = {9, 8, 7, 6, 5},
           C[5], i;

    for (i = 0; i < 5; i++)
        C[i] = A[i] + B[i];

    for (i = 0; i < 5; i++)
        printf(" C[ %d] = %d ", i, C[i]);

    C[0] = A[0] - B[0];    <-- 10 - 9 = 1
    C[1] = A[2] - B[3];    <-- 30 - 6 = 24
    C[2] = 103;

    for (i = 0; i < 5; i++)
        printf(" C[ %d] = %d ", i, C[i]);
}
```

output

c[0] = 19 c[1] = 28 c[2] = 37 c[3] = 46 c[4] = 55  
c[0] = 1 c[1] = 24 c[2] = 103 c[3] = 46 c[4] = 55



#9. (**Boolean logic**) For each of the following code segments indicate what is the output. Assume the following declarations: int w = -3, x = 0, y = 3, z = 6; Note rand() returns a random number.

a.     if (w == -3 && z > 5)  
          printf ("True");  
      else  
          printf("False");

True

b.     if (w\*x == z - y + w)  
          printf ("True");  
      else  
          printf("False");

True

c.     if ( !(w+z == 10 && x == 0) )  
          printf ("True");  
      else  
          printf("False");

True

d.     if ( (x == 0) && (z + y != 9) || y < 2 )  
          printf ("True");  
      else  
          printf("False");

False

e.     if ( y == rand() || x == 0 )  
          printf ("True");  
      else  
          printf("False");

True

f.     z = (x == 0) && (w + y == 0);  
      if ( z )  
          printf ("True");  
      else  
          printf("False");

True

g.     z = (x > 1) || (w == -3);  
      if ( !z )  
          printf ("True");  
      else  
          printf("False");

False



#10. **(Switch statement)** For each of the following code segments indicate what is the output. Assume all variables are declared.

a.     x = 1;  
      switch (x)  
      {  
          case 1:  
              printf ("A");  
          case 2:  
              printf ("B");  
              break;  
          case 3:  
              printf ("C");  
          default:  
              printf ("Z");  
      }

---

AB

b.     x = 4;  
      switch (x)  
      {  
          case 1:  
              printf ("A");  
              break;  
          case 2:  
              printf ("B");  
          default:  
              printf("Z");  
      }

---

Z

c.     x = 2;  
      switch (x)  
      {  
          case 1:  
              printf ("A");  
          case 2:  
              printf ("B");  
              x = 4;  
              break;  
          case 4:  
              printf ("C");  
              break;  
          default:  
              printf("Z");  
      }

---

B

#11. (**Types**) For each of the following code segments indicate what is the output.  
Assume all variables are declared.

- a. `printf (" %d ", 1 / 3);` 0
- b. `printf (" %f ", 1.0 / 3);` 0.333
- c. `printf (" %d ", 1.0 / 3);` 0
- d. `printf (" %f ", 1 / 3);` 0.0
- e. `printf (" %f ", float(1 / 3) );` 0.0
- f. `printf (" %f ", float(1) / 3 );` 0.333
- g. `printf (" %c ", '3' );` 3
- h. `printf (" %d ", '3' );` 49 = 31h