CS2050 –C Programming Quiz 11 SPRING 2017

DO NOT PUT YOUR ANSWERS ON THIS SHEET – RECORD THEM ON THE ANSWER SHEET ONLY.

1. Define a function pointer for a function named “sue” that takes 2 doubles as arguments and returns an integer pointer.

2. T / F One advantage of using function pointers is that they make your programs more easily enhanced or “extended”.

3. Which of the following is a valid way to call a function pointer named fred that takes in an int and a double?  
 a) (\*fred) (3, 4.5);  
 b) fred (3, 4.5);

c) both a & b

d) none of the above

4. T / F Just like an array, a function name is turned into an address when it’s used in an expression.

5. T / F You cannot use the “==” operator to check / compare a function pointer for equality.

Answer the questions below using the knowledge you’ve gained with function pointers. IGNORE COMPILE ERRORS!

x. double sum (double a, double b) { return a + b; }  
y. double sub (double a, double b) { return a – b; }  
z. double mul (double a, double b) { return a \* b; }  
a. double div (double a, double b) { return a / b; }  
b. void operations( )  
c. {  
d. char opt;  
f. double a, b;  
g. double (\*op) (double, double);  
h. while (scanf(“%c %lf %lf”, &opt, &a, &b) != EOF) {  
j. getchar();  
k. if (opt == ‘+’) op = sum;  
m. if (opt == ‘-’) op = sub;  
n. if (opt == ‘\*’) op = mul;  
p. if (opt == ‘/’) op = div;  
r. printf(“%lf\n”, op (a, b) );  
s. } }

7. Which line of code declares a function pointer?

8. Which line of code is an example of “dynamic” or “late” binding?

9. T / F Function pointers are more error-prone than normal pointers.

10. T / F You can pass a function pointer as an argument.

CS2050 – C Programming Quiz 11 NAME \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SPRING 2017 LAB SECTION \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ANSWERS   
  
1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. T / F

3. \_\_\_\_\_\_\_\_\_\_\_\_\_

4. T / F

5. T / F

6. T / F

7. \_\_\_\_\_\_\_\_\_\_\_\_

8. \_\_\_\_\_\_\_\_\_\_\_\_

9. T / F

10. T / F