Introduction to Computer Programming (COP 3223) Section 4 Test #2 Solutions
March 15, 2014

1) (3 pts) Declare E as a constant equal to 2.7182818.

```
#define E 2.71828218
const double E = 2.7182818;
// Grading: either answer is good, 1 pt for each component.
```

2) (6 pts) Write a **single** printf statement that prints out the following:

```
\\\
\"\
\\\
```

```
printf("\\\\\\\n\\\"\\\n\\\\\\\");
// Grading – 1 pt printf, 1 pt "s, 1 pt \\, 1 pt \n, 1 pt \"
// 1 pt ;
```

3) (5 pts) Complete the blank line shown below with an expression so that the code segment below prints out a random integer in between 100 and 199, inclusive.

```
srand(time(0));

int value = 100 + rand()%100 ;
printf("%d", value);
// Grading: 1 pt 100, 1 pt +, 1 pt rand, 1 pt %, 1 pt 100
```

4) (10 pts) The formula for the volume of a cylinder is $V = \pi r^2 h$. Complete the program below so that it asks the user for the volume of a cylinder and its height, calculates its radius and prints it out to 2 decimal places. (Note: The name of the function that takes the square root of a number in the math library is sqrt.)

```
#include <stdio.h>
#include <math.h>
#define PI 3.141592654

int main() {

    double radius, height, volume;
    printf("Please enter the volume and height of the cylinder.");

    scanf("%lf%lf", &volume, & height) ; // 4 pts, 1 for each part
    radius = sqrt(volume/(PI*height)) ; // 5 pts, 1 for each part

    printf("The radius is %.2lf units.\n", radius); // 1 pt
    return 0;
}
```

5) (15 pts) The program written below is indented poorly and has some logical errors in it. Please rewrite it, fixing the logic errors and indenting it appropriately.

```
int main(){ printf("How much money do you have?\n");
int money; scanf("%lf", money);
if (0 <= money <= 100) printf("You’re poor.\n");
else if (101 <= money <= 1000) { printf("Not bad. ");
printf("You're well on your way.\n"); } else if (money > 1000)
printf("You're rich!\n"); else printf("You owe me!\n");
return 0;
}
```

```
int main(){

    printf("How much money do you have?\n");

    int money;
    scanf("%d", &money);

    if (0 <= money && money <= 100)
        printf("You’re poor.\n");

    else if (101 <= money && money <= 1000) {
        printf("Not bad. ");
        printf("You're well on your way.\n");
    }

    else if (money > 1000)
        printf("You're rich!\n");

    else
        printf("You owe me!\n");

    return 0;
}
```

Grading: Indenting = 5 points, grade wholistically
Adding and = 4 pts (2 pts for each)
Either change %d or double – 1 pt
& - 1 pt
Consistent brace placement, correct lines: 4 pts

6) (10 pts) A leap year on Planet X occurs if the year is divisible by 12 or 35, but not 15. (So 24 and 140 are leap years, but 60 is not a leap year.) Complete the program below so that the user enters a year on Planet X and your program determines if that year is a leap year or not.

```
#include <stdio.h>

int main() {

    int year;
    printf("What year do you want to check?\n");
    scanf("%d", &year)

    if ( (year%12 == 0 || year%35 == 0) && year%15 != 0 )
         printf("%d IS a leap year.\n", year);
    else
         printf("%d is NOT a leap year.\n", year);

    return 0;
}
```

Grading: year%12 == 0 2 pts
year%35 == 0 2 pts
|| 2 pts
&& 2 pts
year%15 != 0 2 pts

7) (12 pts) Write a code segment that incorrectly uses a single equal sign when a double equal sign should be used. Explain one test case that produces a correct output anyway and a second test case that produces an incorrect output compared to what the corrected version would produce.

```
int win;
scanf("%d", &win);

if (win = 1)
    printf("You win!\n");
else
    printf("You lose!\n");
```

This code produces the correct output if the user enters 1. It produces incorrect output if the user enters 0. The problem here is that instead of checking for equality, it changes win to 1, which then always evaluates to true.

Grading: 4 pts example, 4 pts correct case, 4 pts incorrect case

8) (20 pts) A ball dropped from a height of h meters falls a total of $4.9t^2$ meters in *t* seconds. Obviously, once it hits the ground, the ball just stays at a height of 0 meters. Complete the program below so that it prints out a chart of the height of a ball at each second requested. Each height should be printed in meters with one digit after the decimal point. For example, if the user enters that the ball starts at a height of 20 meters and we want to print a chart for 5 seconds, the output would be:

```
1    15.1
2    0.4
3    0.0
4    0.0
5    0.0
```

```
#include <stdio.h>
#include <math.h>

#define HALF_ACC 4.9

int main() {

    double starth, curh;
    int numsec, i;

    printf("What height are you dropping the ball?\n");
    scanf("%lf", &starth);
    printf("How many seconds for the chart?\n");
    scanf("%d", &numsec);

    for (i = 1; i <= numsec; i++) {          // 5 pts

        curh = starth - HALF_ACC*i*i;        // 5 pts

        if (curh <= 0)                       // 2 pts
            curh = 0;                        // 3 pts

        printf("%d\t%.1lf\n", i, curh);      // 5 pts
    }


    return 0;
}
```

9) (10 pts) What will the following segment of code print out?

```
int a = 1, b = 3, c;
while (b < 50) {
    c = 2*a + b;
    a = b;
    b = c;
    printf("a = %d, b = %d\n", a, b);
}
```

a = 3, b = 5
a = 5, b = 11
a = 11, b = 21
a = 21, b = 43
a = 43, b = 85

Grading = 1 pt each, grade for cascading errors and only take off for actual error.

10) (8 pts) Write a segment of code that opens a file with the name "odd.txt" and writes to it the first hundred thousand odd positive integers, one per line. The first few lines of the file (after your code segment completes) should look like:

```
1
3
5
```

```
FILE* ifp = fopen("odd.txt", "w");  // 3 pts
int i;
for (i=0; i<100000; i++)             // 2 pts
    fprintf(ifp, "%d\n", 2*i+1);     // 3 pts
fclose(ifp);
```

11) (1 pt) By what acronym is the network "Cable News Network" better known?

CNN (Give to all)