University of Maryland College Park

Dept of Computer Science

CMSC106 Fall 2012

Midterm I Key

Last Name (PRINT): ____

First Name (PRINT): ____

University Directory ID (e.g., umcpturtle)________________________________

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature: ____

Instructions

- This exam is a closed-book and closed-notes exam.
- Total point value is 100 points.
- The exam is a 50 minutes exam.
- Please use a pencil to complete the exam.
- WRITE NEATLY.
- You don't need to use meaningful variable names; however, we expect good indentation.

Grader Use Only

#1	Problem 1 (Miscellaneous)	(40)	
#2	Problem 2 (Conditionals)	(20)	
#3	Problem 3 (Loops)	(20)	
#4	Problem 4 (Loops)	(20)	
Total	Total (100)	(100)	

Problem #1 (40 pts)

1. (2 pts) Circle those identifiers considered valid in C?
   a. ipad√
   b. song√
   c. 34boat
   d. $age
   e. Cmsc++106
   f. #telephone

2. (2 pts) Is the following an infinite loop? Briefly explain (yes or no answers will receive no credit).

```
while(10 = 10) { printf("Here\n"); }
```

   Answer:
   Yes, as 10 = 10 makes the loop expression 10 (which in C is true)

3. (2 pts) The body of a ***do while*** will be executed:
   a. √At least once
   b. At least twice
   c. More than twice
   d. Never
   e. None of the above

4. (2 pts) In a while statement curly brackets are not required if we are executing:
   a. √Only one statement
   b. Two statements
   c. Three statements
   d. None of the above

5. (2 pts) How many values can you represent with 4 bits?

   Answer: 16

6. (2 pts) Write a Unix command that will copy the file public00.in present in the directory **/tmp** to your home directory.

   Answer:
   cp /tmp/public00.in ~

7. (2 pts) Write a Unix command that will compile the program proj1.c.

   Answer:
   gcc proj1.c

8. (2 pts) Which of the following prints the "%" symbol?
   a. √ printf("%%");
   b. printf("%");
   c. printf("\%");
   d. a. and b.
   e. a. and c.
   f. None of the above

9. (2 pts) Which of the following format specifiers skips spaces by default?
   a. %d
   b. %c
   c. %f
   d. a. and b.
   e. √ a. and c.
   f. None of the above

10. (2 pts) Modify the following printf statement (not data) so we can generate the output: **1.2341**

```
float data = 1.234123;
printf("%f", data);
```

Answer:

printf("%.4f", data);

11. (2 pts) Write the output generated by the following code fragment.

```
   int x = 200, y = 4;
   if (x == 200 || ++y)
      printf("%d %d", x, y);
   else
      printf("300\n");
```

Answer:

200 4

12. (2 pts) Rewrite the following code using a single **if** statement.

```
   if (day > 31)
      if (day <= 60)
         printf("February\n");
```

Answer:

```
   if (day > 31 && day <= 60)
      printf("February\n");
```

13. (8 pts) Rewrite the following code using a **switch** statement.

```
if (x == 1) {
   printf("One\n");
} else if (x == 2) {
   printf("Two\n");
} else {
   printf("Other\n");
}
```

Answer:

```
switch(x) {
   case 1:
      printf("One\n");
      break;
   case 2:
      printf("Two\n");
      break;
   default:
      printf("Other\n");
      break;
}
```

14. (2 pts) Write the output generated by the following code fragment if the value of x is 800.

```
if (x < 100)
   if (x < 80)
      printf("Dog\n");
else
   printf("Cat\n");
printf("Done");
```

Answer: Done

15. (2 pts) Rewrite the following statement using a single compound assignment operator.

```
x = x + 1;
x = x + 2;
```

Answer: x += 3

16. (4 pts) Rewrite the following code using a single **while** loop.

```
int x, y;
for (x = 1, y = 2; x < 10; x++, y++) {
   printf("%d %d\n", x, y);
}
```

Answer:

```
int x = 1, y = 2;
while (x < 10) {
   printf("%d %d\n", x, y);
   x++;
   y++;
}
```

Problem #2 (20 pts)

Write a complete C program that computes a student's letter grade based on two exams. The program will read the exam scores, average the values, and generate a message ("A", "B" or "Failed") according to the following cutoffs:

"A" → if the numeric grade is 90 or above
"B" → if the numeric grade is greater than or equal to 70 and less than 90
"Failed" → any value less than 70

For this program:

- You can use the message "Enter value" to read a value.
- You do not need to write pseudocode.
- You do not need to use meaningful variable names.

Answer:

```
#include <stdio.h>

int main() {
   int s1, s2, avg;

   printf("Enter value ");
   scanf("%d", &s1);
   printf("Enter value ");
   scanf("%d", &s2);

   avg = (s1 + s2) / 2;

   if (avg >= 90) {
      printf("A");
   } else if (avg >= 70 && avg < 90) {
      printf("B");
   } else {
      printf("Failed");
   }

   return 0;
}
```

Problem #3 (20 pts)

Write a C program that prints the square roots of even values between 1 and an integer value provided by the user. To compute the square root use the sqrt function (e.g., float val = sqrt(4); will leave 2 in val). Use the message "Enter value: " to read the integer value.

The following is an example that illustrates the functionality of your program when the user enters 7. Notice that your program must work for other scenarios and not just the example below. For this problem you must use a while loop otherwise you will lose significant credit.

```
Enter value: 7
2    1.414214
4    2.000000
6    2.449490
```

Answer:

```
int main() {
   int value, i = 1;

   printf("Enter value: ");
   scanf("%d", &value);
   while(i <= value) {
      if (i % 2 == 0) {
         printf("%d    %f\n", i, sqrt(i));
      }
      i++;
   }

   return 0;
}
```

Problem #4 (20 pts)

Write a program that reads a height (in inches) and displays the height in feet (12 inches in a foot). The program expects a height value between 5 (inclusive) and 96 (inclusive). The program will generate the error message "Invalid height" and keep asking for a height value as long as the user provides an invalid one. Once a correct value has been provided the program will print the height in feet.

The following is an example that illustrates the functionality of your program. Notice that your program must work for other scenarios and not just the example below. Use the message "Enter height: " to read the height. For this problem you must use a **do while** otherwise you will lose significant credit.

```
Enter height: -1
Invalid height
Enter height: 100
Invalid height
Enter height: 63
Height in feets: 5.250000
```

Answer:

```
int main() {
   int height, valid;

   do {
      printf("Enter height: ");
      scanf("%d", &height);
      valid = (height >= 5 && height <= 96);
      if (!valid) {
         printf("Invalid height\n");
      }
   } while(!valid);
   printf("Height in feets: %f\n", height / 12.0);

   return 0;
```