



University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

SEMESTER 2 EXAMINATION – 2011/2012

COMP 10060

Computer Science for Engineers I

Prof. Alain Mille
Mr. J. Dunnion
Prof. J. Murphy*

Time allowed: 2 hours

Instructions for candidates

Answer **Question 1** and **one** other Question.
Question 1 carries 60 marks; Questions 2 and 3 carry 40 marks.

This is a closed-book examination. No calculators allowed

Instructions for Invigilators

This is a closed-book examination. No calculators allowed.

Loose Rough Work sheets are not to be distributed or used.

READ EACH QUESTION CAREFULLY.

Question 1 (COMPULSORY) [60 marks]

Answer all parts (a) – (t). Each part carries 3 marks. No explanation required for any part of Q1

- (a) What is the screen output of the following fragment of C code?

```
int i;
int somearray[] = {3,7,4,6,8};
for (i = 1; i <= 3; i++){
    printf("Element number %d is %d\n", i,
somearray[i]);
}
```

- (b) What is the screen output of the following fragment of C code?

```
int count(int A[], int size, int target) {
    int n = 0, i;
    for (i=0; i<size; i++) {if (A[i]==target)
{n++;}}
    return n;
}

void main()
{
int name[]={5,6,5,3,2,5,6,7,8,2,1,3,4,5};
int result=0;
result = count(name,20,2);
printf("output is %d\n", result);
}
```

- (c) What is the screen output of the following fragment of C code?

```
int i;
for (i=1;i<5;i++){
    switch(i)
    {
        case 1:  printf("1");
        case 3: printf("3\n");
        break;
        case 4: printf("4");
        case 5: printf("5\n");
        default: printf("default\n");
    }
}
```

- (d) What is the screen output of the following fragment of C code?

```
int a=2, b=0, c=-2;
if (a||b&&c){
    printf("False\n");
} else {
    printf("True\n");
}
```

- (e) What is the screen output of the following fragment of C code?

```
int j=10, y;
y = ! j;
printf("y is %d\n",y);
```

- (f) What is the screen output of the following fragment of C code?

```
int nstars = 5, stars;
while (nstars >= 1) {
    stars = 1;
    while (stars <= nstars) {
        printf("+");
        stars++;
    }
    printf("\n");
    nstars--;
}
```

- (g) What is the screen output of the following fragment of C code?

```
int i=-5,j=-i;
if (i<=-10){
    printf("first\n");
} else if ((-i)>=(-j)){
    printf("second\n");
} else {
    printf("no match\n");
}
```

- (h) What is the screen output of the following fragment of C code?

```
double x=42.58;
printf("value is +%2.1f\n",x);
```

- (i) Suppose that x, y, z and w are all of type int. If the initial value of w is 30, x is 20, y is 3, and z is 2, what is the final value of w in the expression?

```
w /= x / y + y * z
```

- (j) What is the screen output of the following fragment of C code?

```
float number = 435.73810, new ;
new = ((int)(10*number))/10.0 ;
printf("number is %7.2f\n",new);
```

- (k) What is the screen output of the following fragment of C code?

```
int i,j=2;
for (i=6;i>2;i--){
    j+=i;
    printf("j is %d\n",j);
}
```

- (l) What is the screen output of the following fragment of C code?

```
double x=3.142;
printf("value is %4.2f",3*x);
```

- (m) What is the screen output of the following fragment of C code?

```
int i=10;
while (i>5){
    i -= 2;
    printf("i is %d\n",i);
}
```

- (n) What is the screen output of the following fragment of C code?

```
#include <stdio.h>
int f1(int a, int b, int c){
    return (b);
}
int f2(int a, int b){
    int c = f1(a, b, 3);
    return c+1;
}
void main()
{
    printf("result is %d\n", f1(-1,f2(2,4),2));
}
```

- (o) What is the screen output of the following fragment of C code?

```
int x = -2;
int y = 3;
int* p = &y;
*p = (*p)*x - (*p)*y;
printf("x is %d and y is %d\n", x, y);
```

- (p) What is the screen output of the following fragment of C code?

```
char str[]="abcdefghijklmn";
char wanted[]="cgdlm";
int i, j;
for (j=0; wanted[j]!='\0'; j++){
    for (i=0; str[i]!='\0'; i++){
        if (str[i]==wanted[j]){
            str[i]='W';
            break;
        }
    }
}
printf("string=%s\n", str);
```

- (q) What is the screen output of the following fragment of C code?

```
struct Address {
    int number;
    char street[20];
    char city[20];
};

struct Employee {
    int number;
    char name[30];
    int age;
    char position[30];
    struct Address addr;
};

void main() {
    struct Employee emp;
    emp.number = 22;
    emp.addr.number = 56;
    printf("output is %d\n", emp.number);
}
```

- (r) What is the screen output of the following fragment of C code? The initial values in datafile.txt are:

0.2 80
0.7 60
0.1 50

```
float prob, avg=0.0;
int quantity, num_values=0;
FILE *fptr;
fptr=fopen("datafile.txt", "r");
while(fscanf(fptr, " %f %d", &prob, &quantity)==2){
/* if return value from this fscanf() is not 2, */
/* the end-of-file indicator has been reached */
avg = avg + (prob*quantity);
}
fclose(fptr);
printf("average value is %.2f\n", avg);
```

- (s) What are the contents of datafile.txt after the execution of this fragment of code? The initial values in datafile.txt are:

200
-54

```
int var1, var2;
FILE *fp;
FILE *fptr;
fp=fopen("datafile.txt", "r");
fscanf(fp, "%d %d", &var1, &var2);
fclose(fp);
fptr=fopen("datafile.txt", "w");
fprintf(fptr, "%d\n%d", var2, var1);
fclose(fptr);
```

- (t) What is the screen output of the following fragment of C code?

```
double x=3.14159;
printf("value is %.3e",x);
```

Question 2 [40 marks]

Answer all parts (a) – (c).

- (a) Consider the following fragment of C code that is using the **while** loop construct:

```
int nstars = 1, stars;
while (nstars <= 10) {
    stars = 1;
    while (stars<=nstars) {
        printf("*");
        stars++;
    }
    printf("\n");
    nstars++;
}
```

Re-write this code fragment using **for** loops instead of **while** loops.

[10 Marks]

- (b) Consider the following C program:

```
#include <stdio.h>
void main()
{
    FILE *f_ptr_in, *f_ptr_out;
    int inp;
    f_ptr_in = fopen("input.dat", "r");
    f_ptr_out = fopen("output.dat", "w");
    while (fscanf(f_ptr_in, "%d", &inp)==1){
        if (inp < 10 && inp > -10){
            fprintf(f_ptr_out, "%d\n", inp);
        } else
            fprintf(f_ptr_out, "out of bounds\n");
    }
    fclose(f_ptr_in);
    fclose(f_ptr_out);
}
```

- i. Suppose the file **input.dat** contains the following data:

```
0
12
-13
9
33
8
2
-4
```

After executing this program, what does the file **output.dat** contain?

[10 Marks]

- ii. Re-write the above program so that it writes all the even values in the file **input.dat** to a file called **output2.dat**.

[10 Marks]

(c) Consider the following C program:

```
#include "stdio.h"
void main(void) {
    char message[80]="I love C programming";
    int i=0, count=0;
    while (message[i]!='\0') {
        if ((message[i]=='c') || (message[i]=='C')) {
            count++;
        }
        i++;
    }
    printf("\n%s\n contains %d c's\n", message, count);
}
```

Re-write the above program so that all the code for determining the number of **c**'s in the string **message** is contained in a function called **c_counter()** that you should define, while the declaration and initialization of **message** and the output is still done from **main()**.

[10 Marks]

Question 3 [40 Marks]

Answer both parts (a) and (b).

(a) Consider the following C program:

```
#include "stdio.h"
void main(void){
    char message[80]="hello there";
    int count=0;
    while (message[count]!='\0'){
        count++; /* increase by 1 as long as NULL not reached */
    }
    printf("\"%s\" has length %d\n", message, count);
}
```

(i) What is the screen output of this program (*no explanation required*)?

(ii) **Re-write** the above program so that all the code for determining the length of the string **message** is contained in a function called **stringlength()** that you should define.

[20 Marks]

(b) Write a program that uses a function **convert()** to determine the equivalent number of hours, minutes, and seconds for a given time in seconds. For example, 3,661 seconds is equivalent to 1 hour, 1 minute, and 1 second. Use the following function prototype:

```
void convert(int time, int *phrs, int *pmins, int *psecs);
```

Remember to use good programming standards, define the problem statement, show the design of the program, add comments to the code, and follow good formatting practices to lay out the code. Your answer should include the algorithm design and the code.

[20 Marks]