University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

SEMESTER 2 EXAMINATION – 2005/2006

FIRST EXAMINATION IN ENGINEERING

Computer Science for Engineers

COMP 10060

Prof. J. Kramer
Prof. B. Smyth
Dr. J. Murphy*

Time allowed: 2 hours

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.

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.

(a) True or False *(no explanation required)*: in a C program, comments **must** be included to explain and document what the program does, and why.

(b) True or False *(no explanation required)*: a logic error in a C program will be **detected** by the compiler, although the error message may be difficult to understand.

(c) If `x=5`, `y=4`, and `z=3`, what is the value of `w` in the expression `w=x+y/z` ?

(d) What is the screen output of the following fragment of C code *(no explanation required)*:

```
float x=2.23607;
printf("value is %.4f\n",x);
```

(e) What is the screen output of the following fragment of C code *(no explanation required)*:

```
int i=-1,j=2;
if (j<=i){
  printf("normal case\n");
} else if ((-i)>(j/2)){
   printf("exceptional case\n");
  } else {
     printf("error case\n");
    }
```

(f) What is the screen output of the following fragment of C code *(no explanation required)*:

```
int a=0, b=2, c=-1;
if ((!a) && ((b*c)>0) ){
  printf("this one\n");
} else {
     printf("that one\n");
   }
```

(g) What is the screen output of the following fragment of C code *(no explanation required)*:

```
int i;
for (i=21;i>0;i--){
  i = i/3;
  printf("i is %d\n",i);
}
```

(h) Select the correct answer: The function prototype

```
float fname(int *x);
```

tells us that

(h-1) `fname()` takes 1 argument of type pointer-to-int and returns a value of type float.
(*h-2)* `fname()` takes 1 argument of type int and returns a value of type float.
(h-3) `fname()` takes 1 argument of type float and returns a value of type pointer-to-int.

[Question 1 continues]

Question 1 (continued)

(i) What is the screen output of the following fragment of C code *(no explanation required)*:

```
int i;
int array[5] = {1,2,3};
for (i=0; i<=4; i++){
  printf("element number %d is %d\n", i+1, array[i]);
}
```

(j) Select the correct answer: If the pointer **ptr1** currently points to **x** and you want to assign the current value of **x** to **y**, you could use the statement

(j-1) **ptr1 = &y;**
(j-2) ***ptr1 = y;**
(j-3) **y = *ptr1;**

(k) What is the screen output of the following fragment of C code *(no explanation required)*:

```
int x = 1;
int y = -2;
int* p = &x;
*p = (*p)*y + (*p)*x;
printf("x is %d and y is %d\n", x, y);
```

(l) What is the screen output of the following fragment of C code *(no explanation required)*:

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

(m) Select the correct answer: **fopen("datafile.txt","r")** means

(m-1) if it exists, open **datafile.txt** for random access, otherwise return an error.
(m-2) if it exists, open **datafile.txt** for reading only, otherwise return an error.
(m-3) if it exists, open **datafile.txt** for reading only, otherwise open the first file found and read from it.

(n) Complete the sentence: The two most widely used tools for developing algorithms are flowcharts and _____ .

[Question 1 continues]

Question 1 (continued)

(o) What is the screen output of the following fragment of C code *(no explanation required)*:

```
int i;
for (i=0;i<=4;i++){
  switch(i){
    case 1:  break;
    case 2:  printf("2\n");
             break;
    case 3:  printf("3\n");
    default: printf("default\n");
             break;
  }
}
```

(p) What is the screen output of the following C program *(no explanation required)*:

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

(q) What is the screen output of the following fragment of C code *(no explanation required)*:

```
double x[3];
int i, y[3];
for (i=0; i<3; i++){
   x[i] = 2.0/(i+1);
   y[i] = x[i];
}
printf("x[0] is %.2f\n", x[0]);
printf("x[1] is %.2f\n", x[1]);
printf("x[2] is %.2f\n", x[2]);
printf("y[0] is %i\n", y[0]);
printf("y[1] is %i\n", y[1]);
printf("y[2] is %i\n", y[2]);
```

(r) What is the screen output of the following fragment of C code *(no explanation required)*:

```
int i;
char str[]="I hope to do well on this Exam.";
for (i=0; str[i+1]!='\0'; i++){
  if ((str[i]==' ') && (str[i+1] >= 'a') && (str[i+1] <= 'z')) {
    str[i+1] += 'A' - 'a';
  }
}
printf("str=%s\n", str);
```

[Question 1 continues]

Question 1 (continued)

(s) What is the screen output of the following fragment of C code *(no explanation required)*:

```
double x[3] = {1.5, 2.2, 4.3};
printf("value is %.2f\n", *(x+1) );
```

(t) Given the following definition and declaration:

```
struct Employee {
  int number;
  char name[30];
  int age;
  char position[30];
};
struct Employee emp1;
```

Which of the following statements correctly assigns the value **47** to **emp1**'s age?

(t-1) `emp1.age = 47;`
(t-2) `emp1->age = 47;`
(t-3) `emp1-age = 47;`

Question 2 [40 marks]

Answer parts (a) and (b).

(a) 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.

(b) Write a program to compute the received Doppler frequency of a radar echo for a target moving towards the transmitter at velocities ranging from 10 m/s to 100 m/s in 10 m/s steps. The received Doppler frequency Fr is given by:

(Fr – Ft)/Ft=2V/c
Where V is the velocity in m/s
Ft=5.5 GHz the transmitted frequency
c=3 10^8 m/s

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.

Question 3 [40 marks]
Answer all parts (a) – (c).

(a) Consider the following C program:

```
#include <stdio.h>
/* DEFINITION OF FUNCTION "numneg" GOES HERE */
void main(void) {
  int i, j=0, array1[8]={1,-1,-1,1,0,1,0,-1};
  j = numneg(array1,8); /* function call to numneg() */
  printf("there are %d negative elements of array1[]", j);
}
```

Write down the definition of function **numneg()** which counts the number of negative elements in its input array, so that the output of the above program is:

```
there are 3 negative elements of array1[]
```

(b) Consider the following C program:

```
#include <stdio.h>
void main(void)
{
    int intarr[4], i, total=0;
    for (i=0 ; i<4; i++){
       printf("enter value number %d: ", i+1);
       scanf("%d", &intarr[i]);            /* LINE 1 */
       total += intarr[i];                 /* LINE 2 */
          }
    printf("sum of inputs is %d, first input was %d, last
input was %d\n", total, intarr[0], intarr[3]);  /* LINE 3 */
}
```

Re-write the lines **LINE 1**, **LINE 2**, and **LINE 3** using "array pointers" instead of array subscripts.

(c) Consider the following C program:

```
#include <stdio.h>
void main(void)
{
   FILE *fptr1, *fptr2;
   int inp;
   fptr1 = fopen("input.dat", "r");
   fptr2 = fopen("output.dat", "w");
   while (fscanf(fptr1, "%d", &inp)==1){
      if (inp >= 0){
         fprintf(fptr2, "%d ", inp);
      }
   }
   fclose(fptr1);
   fclose(fptr2);
}
```

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

```
0    -2    2    6    -1    -5
```

Complete this sentence: After executing this program, the file **output.dat** contains __

(ii) ***Re-write*** the above program so that it writes all values between **-1** and **4** (inclusive) in the file **input.dat** to a new file called **output2.dat**

o O o