

UNIVERSITY OF PORTSMOUTH

FACULTY OF TECHNOLOGY

Department of Electronic and Computer Engineering

**B142L – Introduction to Computing**

U13746

Date: 14 July 2008

Time: 11:15 – 13:15

**INSTRUCTIONS**

**Write your student ID number clearly on page 2.**

Write your answers to all 10 questions within the spaces provided in this examination paper.

**Handwritten notes are permitted with this examination.**

**Calculators permitted are:**

Casio FX 85WA

Casio FX 83WA

Casio FX 85MS

**Examiner:**

Professor Ron Pethig, Chi Nguyen

**Student ID Number**

**\*\*\* SOLUTIONS \*\*\***

## QUESTION 1

Modify the following source code to correct all errors. Line numbers have been provided for reference. Indicate specific line numbers in your answers when necessary. [10 Marks]

```
01: int main( void )
02: {
03:     int "car";
04:     double mileage;
05:     char summary{ 64 } = '\0';
06:
07:     car = "5";
08:     mileage = 250.70;
09:     sprintf( summary, "%c %m", car mileage );
10:     printf( "%s\n", summary );
11: }
```

```
int main( void )
{
    int car; [1]
    double mileage;
    char summary[ 64 ] = {'\0'}; [4]

    car = 5; [1]
    mileage = 250.70;
    sprintf( summary, "%d %lf", car, mileage ); [4]
    printf( "%s\n", summary );
}
```

**QUESTION 2**

a) Place an "X" in the box next to **3 terms** that are most directly related to the use of **selection and decisions** in a C program. [3 Marks]

	variable	1	if
	index		type
1	else		local
1	and		return
	int		parameter

b) Write a C program that **accepts an input number** and displays a message that it is a **valid** (e.g. **1 to 12**) **or invalid month number**. Show or use all the terms selected in part (a). [7 Marks]

*Similar to:*

```

int main( void )
{
    int month;

    printf( "Please enter a month number: " );
    scanf( "%d", &month );

    if ( ( month >= 1 ) && ( month <= 12 ) )           /* and logic */
        printf( "A valid month" );
    else
        printf( "Not a valid month" );
}
    
```

} [2]  
 } [5]

### QUESTION 3

**Modify the following source code to correct all errors.** The program is intended to **display numbers in the range of 40 to 500 (inclusive) which are multiples of both 5 and 6.** Line numbers have been provided for reference. Indicate specific line numbers in your answers when necessary. [10 Marks]

```
01: int main( void )
02: {
03:     int counter = 0;
04:
05:     do
06:     {
07:         if ( ( ( counter % 5 ) == 0 ) || ( ( counter % 6 ) == 0 ) )
08:             printf( "%d ", counter % 5 );
09:
10:         counter = counter + 6;
11:     } while ( counter < 500 );
12: }
```

*Similar to:*

```
int main( void )
{
    int counter = 40; [2]

    do
    {
        if ( ( ( counter % 5 ) == 0 ) && ( ( counter % 6 ) == 0 ) ) [2]
            printf( "%d ", counter ); [2]

        counter++; [2]
    } while ( counter <= 500 ); [2]
}
```

**QUESTION 4**

a) Place an "X" in the box next to **3 terms** that are most directly related to the use of **iterative loops** in a C program. [3 Marks]

	key
1	body
	printf
	stderr
	element

1	for
	void
1	initialization
	allocation
	format

b) Write a C program that uses **an iterative loop to display all multiples of 3 starting at 600 and decreasing to zero**. Show or use all the terms selected in part (a). [7 Marks]

*Similar to:*

```
int main( void )
{
    int number;
} [1]

/* number is assigned the value 600 at initialization */
for ( number = 600; number >= 0; number = number - 3 ) [4]
    printf( "%d ", number ); /* body of loop */ [2]
}
```

### QUESTION 5

Consider the following source code. Line numbers have been provided for reference. Indicate specific line numbers in your answers when necessary.

```
01: void color( int );
02:
03: int main( void )
04: {
05:     int number = 1;
06:
07:     while ( number != 0 )
08:     {
09:         scanf( "%d", &number );
10:         color( number );
11:     }
12: }
13:
14: void color( int input )
15: {
16:     if ( input >= 10 )
17:         printf( "Red " );
18:     if ( input >= 20 )
19:         printf( "Blue " );
20:     if ( input <= 30 )
21:         printf( "Green " );
22:     if ( input <= 40 )
23:         printf( "Orange " );
24: }
```

a) List the program output when the **input value is 20**.

[3 Marks]

Red Blue Green Orange

[3]

b) Describe **input values** that would produce **output of "Green Orange"**.

[7 Marks]

Integer values **equal to**

[2]

or **less than 9**.

[5]

## QUESTION 6

Consider the following source code. Line numbers have been provided for reference. Indicate specific line numbers in your answers when necessary.

```
01: int main( void )
02: {
03:     int counter;
04:     int marks[ 12 ] = { 0 };
05: }
```

a) Modify the source code to **accept and store 12 input marks in the array.** [4 Marks]

*Similar to:*

```
int main( void )
{
    int counter;
    int marks[ 12 ] = { 0 };

    for ( counter = 0; counter < 12; counter++ )           [2]
    {
        printf( "Please enter a mark: " );
        scanf( "%d", &marks[ counter ] );                 } [2]
    }
}
```

b) Modify the source code to **accept and store 12 input marks in the array only if the input values are within the range of 0 to 100, inclusive.** [6 Marks]

*Similar to:*

```
int main( void )
{
    int counter;
    int marks[ 12 ] = { 0 };

    for ( counter = 0; counter < 12; counter++ )
    {
        do
        {
            printf( "Please enter a mark: " );
            scanf( "%d", & marks[ counter ] );
        }
        while ( ( marks[ counter ] < 0 ) || ( marks[ counter ] > 100 ) ); [6]
    }
}
```

**QUESTION 7**

a) Place an "X" in the box next to **3 terms** that are most directly related to the use of **text strings** in a C program. [3 Marks]

	cast
	true
1	char
1	length
	binary

	assignment
	default
	double
	sizeof
1	array

b) Write a C program that **accepts an input text string** and **displays it to screen except for the letter 'z'**. Show or use all the terms selected in part (a). [7 Marks]

```
Similar to:

#include <string.h>

int main( void )
{
    char text[ 256 ] = { '\0' };          /* array length is 255 */ [2]
    int counter;

    printf( "Please enter a string: " );
    scanf( "%s", text );                 [1]

    for ( counter = 0; counter < strlen( text ); counter++ ) [1]
    {
        if ( text[ counter ] != 'z' )    [2]
            printf( "%c", text[ counter ] ); [1]
    }
}
```

### QUESTION 8

Write a C program that **accepts 5 input text strings** and **displays a message to indicate whether there are duplicate input strings**.

[10 Marks]

*Similar to:*

```
#include <string.h> [1]

int main( void )
{
    char text[ 5 ][ 256 ] = { '\0' }; [2]
    int counter1, counter2;

    for ( counter1 = 0; counter1 < 5; counter1++ )
    {
        printf( "Please enter a string: " );
        scanf( "%s", text[ counter1 ] );
    } [2]

    for ( counter1 = 0; counter1 < 5; counter1++ )
    {
        for ( counter2 = 0; counter2 < 5; counter2++ )
        {
            if ( counter2 != counter1 )
            {
                if ( strcmp( text[ counter2 ], text[ counter1 ] ) == 0 )
                {
                    printf( "Duplicate string found\n" );
                    counter1 = counter2 = 5;
                }
            }
        }
    } [5]
}
```

**QUESTION 9**

a) Place an "X" in the box next to **3 terms** that are most directly related to the use of **data files** in a C program. [3 Marks]

	stdin
	stdout
	stderr
	file array
1	file pointer

1	end of file
	start of file
	scan mode
1	read mode
	delete mode

b) Write a C program that **saves the string "Hello world"** to a **data file named "output.txt"** located in the same directory. Show or use all the terms selected in part (a). [7 Marks]

*Similar to:*

```
#include <stdio.h> [1]

int main( void )
{
    FILE* output = fopen( "output.txt", "w" ); [2]

    if ( output != NULL ) [1]
    {
        fprintf( output, "Hello world\n" ); [2]
        fclose( output ); [1]
    }
}
```

## QUESTION 10

Consider the following source code. Line numbers have been provided for reference. Indicate specific line numbers in your answers when necessary.

```
01: double kilometers( double );
02:
03: int main( void )
04: {
05: }
```

- a) Modify the source code to **define the kilometers() function** that converts **input number of miles** into an **equivalent number of kilometers**. Use this formula:

$$1 \text{ mile} = 1.609 \text{ kilometer}$$

[5 Marks]

*Similar to:*

```
double kilometers( double );
```

```
int main( void )
{
}
```

```
double kilometers( double miles )
```

[2]

```
{
    return ( miles * 1.609 );
}
```

[3]

- b) Modify the source code to **accept an input number of miles** and **display the output of the kilometers() function**.

[5 Marks]

*Similar to:*

```
double kilometers( double );
```

```
int main( void )
{
```

```
    double input;
```

[1]

```
    printf( "Please enter miles for conversion: " );
```

```
    scanf( "%lf", &input );
```

[2]

```
    printf( "That is %lf kilometers\n", kilometers( input ) );
```

[2]

```
}
```

```
double kilometers( double miles )
```

```
{
    return ( miles * 1.609 );
}
```