(National Council for Vocational Awards)

MARKING SCHEME & SOLUTIONS

Computer Programming C20013

Theory Examination 2006

Duration: Two Hours

INSTRUCTIONS TO CANDIDATES:

*Answer any **three** questions*

All questions carry equal marks

Answer the questions using the spaces in this exam booklet

Return this question & answer paper when finished

This written exam counts as 40% of the total module

NAME (PRINT):_____

PPS NUMBER:_____ **MARKING SCHEME**

DATE:_____ **& SOLUTIONS**

Question 1. Total 40 marks.

(a) This program contains 4 errors that will stop it from compiling. List the errors. *20 marks* 5 each

```
include <stdio.h>
int loopy;
main ()
{
 looper = 1;
 printf ("+--------+\n");
 while (loopy <= 8)
 {
  printf ("|         |\n);
  loopy+;
 }
 printf ("+--------+\n");
}
```

1	*#include <stdio.h>*
2	loopy *vs* looper used as variable name
3	Closure of inverted commas in printf
4	loopy+ instead of loopy++

(b) What is a variable used for? *10 marks*

To store data as the program runs, as well as to allow that data to be changed as required.

(c) What is the difference between an integer (**int**) and a float (**float**) variable? Give a sample of each type of data. *10 marks*

One stores integer data – whole numbers (such as 2 or 56 or -4563); the other stores decimal data such as -1.02 or 45.9 or 2345.67

Question 2. Total 40 marks.

(b) Write the general form of the *if...else* statement: **10 marks**

```
if (condition)
{
 action when condition is true;
}
else
{
 action when condition is false;
}
```

(b) Write the general form of the **while** statement: *10 marks*

```
initialize control variable;
while (terminating condition not met, based on control variable)
{
 loop action – the function of the loop;
 progress towards loop termination, based on control variable change;
}
```

(c) The following **C** code will compile and run but will not generate the desired output. Why?

20 marks

```
#include <stdio.h>
// A sample program.
// This program should write out the letters a..z
// of the alphabet, one on each line.
int controlvar;
char alpha; this variable has no function or effect – a distraction!
main ()
{
 controlvar = 97;
 while (controlvar <= 122);
 {
  // This line converts and then writes the character
  printf ("%c\n", controlvar);
  controlvar++;
 }
}
```

The while statement has a semi-colon at the end that will cause an infinite loop.

Question 3. Total 40 marks.

(a) Indicate the values in each of the variables **a**, **b** and **c** after this program finishes:

```c
#include <stdio.h>
main ()
{
 int a, b, c, lv;
 lv = 1;
 // Don't loop when lv is ten
 while (lv != 10)
 {
  a = lv;
  b = lv * 2;
  // Add 1 to lv each time
  lv++;
 }
 c = a * b;
}
```

Variable	Value
a	9
b	18
c	162

(b) What screen output is generated by this program line: *10 marks*
```c
printf ("%c%c%c%c%c\n%c%c%c%c%c%c\n",
68,111,110,39,116,80,97,110,105,99,33);
```

Dont,

Panic!

Question 4. Total 40 marks.

(a) Write a C loop to read in an array of 30 numeric variables and then write out the total (sum) and the average of the numbers which have been read in.

30 marks

```
'Program' not specified to be written -  just a loop; therefore assume the variables are
declared and any other program essentials are done.

total = 0;
control =  1;
while (control <=  30)
{
 printf ("Please enter a number: ");
 scanf ("%d", &anum);
 total = total + anum;
 control++;
}
average = total / 30;
printf ("The total of the numbers is: %d and the average is: %d",
total, average);
```

(b) The control variable for a **while** loop should appear in a program not less than four times. List those times. *10 marks* 2.5 each

1	declare
2	initialize
3	compare (for termination)
4	progress (ie increment, decrement)

Figure 1. The ASCII table.

		032	SP	033	!	034	"	035	#		
036	$	37.00%		038	&	039	'	040	(	041	)
042	*	043	+	044	,	045	–	046	.	047	/
048	0	049	1	050	2	051	3	052	4	053	5
054	6	055	7	056	8	057	9	058	:	059	;
060	<	061	=	062	>	063	?	064	@	065	A
066	B	067	C	068	D	069	E	070	F	071	G
072	H	073	I	074	J	075	K	076	L	077	M
078	N	079	O	080	P	081	Q	082	R	083	S
084	T	085	U	086	V	087	W	088	X	089	Y
090	Z	091	[	092	\	093	]	094	^	095	_
096	`	097	a	098	b	099	c	100	d	101	e
102	f	103	g	104	h	105	i	106	j	107	k
108	l	109	m	110	n	111	o	112	p	113	q
114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124			125	}
126	~	127									
Printable alphanumeric and punctuation characters used in normal document text											

Rough-Work Page

Rough-Work Page