FETAC
Further Education and Training Awards Council
Comhairle na nDámhachtainí Breisoideachais agus Oiliúna

(National Council for Vocational Awards)

Computer Programming C20013

Theory Examination 2005

Duration: Two Hours

INSTRUCTIONS TO CANDIDATES:

*Answer any **three** questions*

All questions carry equal marks

Return this exam/answer paper when finished

Extra paper is available from the exam supervisor if required

This written exam counts as 40% of the total module

NAME (PRINT): **Worked Solution**

PPS NUMBER: _____

DATE: _____

Question 1. Total 40 marks.

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

```
#INCLUDE <stdio.h>
main ()
int count;
cher one_letter;
one_letter = ' ';
printf (“Enter characters. Stop by pressing a full stop.”):
while (one_letter != '.')
{
 printf (“%c \n”, one-letter);
 printf (“Next character, please: ”);
 scanf (“%c”, &one_letter);
}
}
```

1	#INCLUDE should be lowercase
2	Missing { after main – leads to extra (but required }) at end of program.
3	Use of : instead of ; on line 6
4	char mis-spelled as cher
5	one-letter on line 9 should be one_letter

(b) What kind of data is stored in a float variable? Why can't int be used in this case? ***10 marks***

Decimal data should be stored in a float variable. Int cannot store the fractional/decimal part – it's designed fro whole numbers only.

(c) What is the difference between a variable and a constant? ***10 marks***

A variable is used to store changeable data in a program. Constants, as the name implies, store data that does not change.

Question 2. Total 40 marks.

(a) How many brackets and inverted commas should be used in a C computer program? ***10 marks***

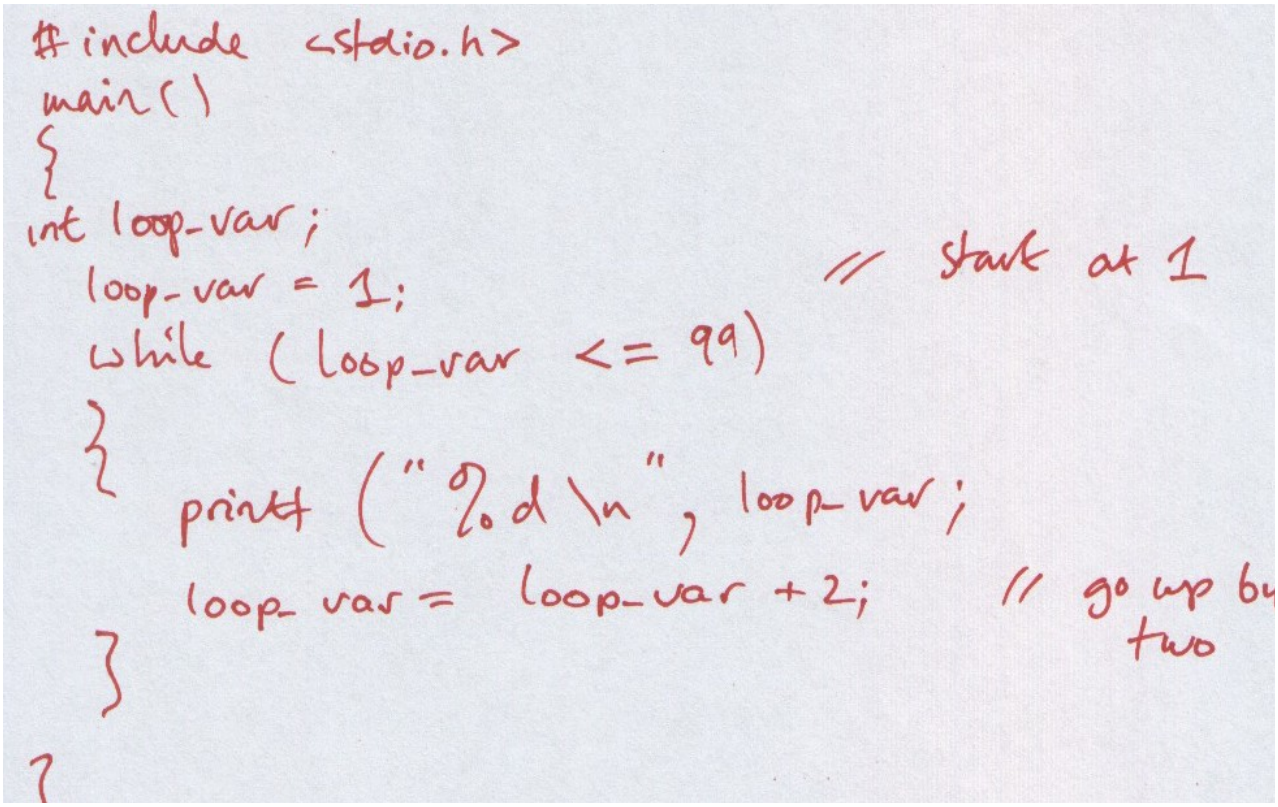
An even number of each. Every left bracket should be matched by a right bracket and the same rule apples for inverted commas.

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

```
if (condition)
{
 action1  // condition is true
}
else
{
 action2 // condition is false
}
```

(c) Write a C program containing a loop that writes out the odd numbers between 9 and 99

20 marks



Question 3. Total 40 marks.

(a) **What output will the following program generate on screen?** ***30 marks***

```
#include <stdio.h>
#define start_symbol 58
int looper;
char thesymbol;
main ()
{
 thesymbol = start_symbol;
 looper = 1;
 while (looper <= 5)
 {
  thesymbol = thesymbol + 1;
  printf ("%c", thesymbol);
  thesymbol = thesymbol - 14;
  printf ("%c", thesymbol);
  thesymbol = thesymbol + 34;
  printf ("%c", thesymbol);
  printf ("\n");
  thesymbol = thesymbol - 21;
  looper++;
 }
}
```

Write the output here:

;-O
;-O
;-O
;-O
;-O

(b) What screen output is generated by this program line: ***10 marks***

```
printf ("%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n", 82, 111, 121, 32, 75, 101,
97, 110, 101, 32, 119, 97, 115, 32, 114, 105, 103, 104, 116, 33);
```

Roy Keane was right!

Question 4. Total 40 marks.

(a) What output will be generated by this program:

```
#include <stdio.h>
main ()
{
 int anumber, loopvar, total;
 loopvar = 0;
 total = 0;
 while (loopvar <= 9)
 {
  anumber = 200 - (loopvar * loopvar);
  if ((loopvar == 2) || (loopvar == 6) )
  {
   anumber = 0;
  }
  printf ("%d\n", anumber);
  loopvar++;
  total = total + anumber;
 }
 printf ("Total calculated: %d\n", total);
}
```

30 marks

Write the expected output here:

```
200
199
0
191
184
175
0
151
136
119
Total calculated: 1355
```

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

1	Declare
2	Initialize
3	Compare
4	Progress (towards end of loop)

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