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

(National Council for Vocational Awards)

Computer Programming C20013

Sample Theory Examination 2007

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 sample written exam counts as 40% of the total module

NAME (PRINT): SAMPLE

PPS NUMBER: _____

DATE: _____

Question 1. Total 40 marks.

(a) This program contains 4 errors that will stop it from compiling. List the errors.

20 marks

```
!/usr/bin/perl
$myname = "John"
$n = 1;
while ($n <= 4)
{
  print ("$myname, $n\n");
  $n=$n++1;
}
```

1	Missing #! at start
2	Missing ; at end of line
3	$n++1; incorrect should be $n++;
4	Incorrect type of matching bracket

(b) What are variables used for? ***10 marks***

To store data during the running of a program.

(c) What is the difference between a variable with a name starting with **$** and a variable starting with @? ***10 marks***

a $ means the variable is a 'scalar' with one value. A @ means the variable is an 'array' with several values.

Question 2. Total 40 marks.

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

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

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

```
Set control_variable value
while ( test control_variable for
          time to end the loop )
{
   ======  <-- do some stuff
   ======
   change value of control variable
          towards an ending value
}
```

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

20 marks

```
#!/usr/bin/perl
# A sample program.
# This program should write out the letters a..z
# of the alphabet, one on each line.
$startvar = 97;
$stopvar = 122;
$counter = $startvar;
while ($counter <= $stopvar)
{
  # This next line converts & prints the character
  printf ("%c\n", $counter);
  $counter--;
}
```

The loop counts down (--) not up (++) as required.

Question 3. Total 40 marks.

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

30 marks

```
#!/usr/bin/perl
$num = 0;
$a = $num * 2;
while ($num <= 9)
{
  $num=$num+1;
}
$b = 12;
$c = $num * 2;

print ("$a, $b, $c\n");
```

the end value of a loop variable is always 1 bigger! $num will end at 10

Variable	*Value*
$a	0
$b	12
$c	20

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

```
#!/usr/bin/perl
printf ("%c%c%c%c%c%c%c%c%c%c%c%c\n",
68,111,110,39,116,80,97,110,105,99,33);
```

Don't Panic!

Question 4. Total 40 marks.

(a) Write a perl loop to write out every second number between 1 and 30 and then write out the the average of the numbers which have been printed in.

30 marks

```
$average = 0;     $count = 1;
$counter = 1;
while ( $counter <= 30)
{  print " counter\n";
 $average = $average + $counter;
$count ++; # an extra number
 $counter = $counter + 2;  # jump the even nos.
}
$average = $average / $count
# divide the total of the numbers by the
# count of the numbers for the average
print "\n Average = $average\n";
```

(b) In many languages the control variable for a **while** loop should appear in a program not less than four times. In perl, only 3 occurrences are required. List those times.

10 marks

1	Setting the first value of the control variable
2	Testing the value of the control variable
3	Changing the control variable to progress towards the end state of the 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