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

Computer Programming
C20013
Solutions

Theory Examination 2008
This written exam counts as 40% of the total module

Duration: Two Hours

INSTRUCTIONS TO CANDIDATES:

1. Answer any **three** questions
2. All questions carry equal marks
3. Return this exam paper when finished along with your answer book
4. Answer the questions using the spaces in this exam booklet

Candidate Name: _____ **Date:** _____

PPS Number: _____

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
print "Please enter the first number: ";
$first = <STDIN>;
chop $first;
print "Now enter the second number: ";
$second = <STDIN>;
chop $second;
$result = $first * $second;
print "The product of the numbers is: $result.\n";
```

1	
2	
3	
4	

(b) What is the difference between a variable prefixed with **$** and one prefixed with the @ symbol?

10 marks

A variable commencing with $ is a scalar variable that contains one value, the @ indicates an array with multiple values.

(c) In the array **@parts** what number would replace **y** to represent the *last* element in an array of 10 elements?

10 marks

$parts[y] **y = ?___9___?**

Question 2. *Total 40 marks.*

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

```
initialize condition for starting
while (end condition not reached)
{
   do something useful in the loop
   progress towards the end condition
}
```

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

```
if (condition is true)
{
   do something because the condition is true
}
else
{
   do something else because the condition is false
}
```

(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/formats & prints the character
  printf ("%c\n", $counter);
  $counter--;
}
```

```
# Should be $counter++ to count upwards
```

Question 3. *Total 40 marks.*

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

3 x 10 marks

```
#!/usr/bin/perl
print "Content-type: text/html\n";
print "<html><body>\n";
$num = 0;
$a = 2 * 2 * 2 * 2;
while ($num <= 5)
{
  print "<br>$num";
  $c = $num * 5;
  $num=$num+1;
}
$b = $num/2;
$c = $c * 2;
print ("<hr>\n");
print ("<br>A=$a,<br>B=$b,<br>C=$c\n");
print ("</body></html>");
```

Variable	*Value*
$a	16
$b	3
$c	50

(b) What screen output is generated by this short program using the *printf* command:

10 marks

```
#!/usr/bin/perl
printf ("%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c\n",
89,111,117,039,108,108,32,110,101,118,101,114,32,119,97,108,
107,32,97,103,97,105,110,33);
```

You'll never walk again!

Question 4. *Total 40 marks.*

(a) Write a perl web script containing a loop to write out every number from 2 to 20 and then write out the total of all the numbers which have been displayed. Each item should be on a separate line in the web page generated.

20 marks

One possible solution (with comments) is:

```
#!/usr/bin/perl
# This line is so the web server knows what to do
print "Content-type: text/html\n";
print "<html><body>";
$total=0;
$mycounter=2;
$stop=20;
# Now start printing and adding numbers
while ($mycounter <= $stop)
{
  print "<br />$mycounter";
  # Keep a running total of the numbers...
  $total = $total + $mycounter;
  $mycounter++;
}
print "<br />The total is: $total";
print "<br /></body</html>";
# Ta-Dah! End of script.
```

(b) Write a short program with a **while** loop that repeatedly asks for numbers and writes out the square (*i.e.* the product of the number by itself) of each number. The loop should stop when it receives the value **-1**.

20 marks

One possible solution is:

```
#!/usr/bin/perl
$numnum=+1;
while ($numnum != -1)
{
  print "Please enter a number: ";
  $numnum = <STDIN>;
  $square = $numnum * $numnum;
  print "The square of that number is $square\n";
}
```

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