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

Sample Solutions
Computer Programming C20013

May 2011

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): **Sample Solutions**

PPS NUMBER: _____

DATE: **2011**

Question 1. Total 40 marks.

(a) This program contains 6 errors that will stop it from compiling. List the errors.

6 * 5 marks

```
#!/user/bin/perl
$first name = "";
$loop = 1;
while ($loop <= 3)
{
  Print "Type your name; ":
  $first_name = <STDIN>;
  if ($first_name eq "Bill") {
    print "You are Bill";
  }
  elseif ($first_name eq "Adam") {
    print "You are Adam";
  }
  else
    print "Hello, $forst_name";
  }
  $loop=$loop+1;
}
```

1	**$first name = ""; # should be $first_name = "";**
2	**Print "Type your name; ": # should be print**
3	**Print "Type your name; ": # should end with ;**
4	**elseif ($first_name eq "Adam") { # should be elsif**
5	**else # missing opening {**
6	**#!/user/bin/perl # should be #!/usr/bin/perl**
X	**print "Hello, $forst_name"; # this is a trap. It's a semantic error – the program will still compile!**

(b) There is an error in this code. Is it a *syntax* error or a *semantic* error? ***10 marks***

```
$data = <STDIN>;
if ($data = 10) { SEMANTIC: should be == for comparison
  print "Ten."; }
```

Question 2. Total 40 marks.

(a) Write the general form of the ***if...elsif...else*** *statement:* ***15 marks***

```
if (condition1) {
  perform action 1 #  because condition1 is true
}
elsif (condition2) {
  perform action 2 # because condition1 is false but condition2 is
true
}
else {
  perform action 3 # the default action when both condition1 and
condition2 are false
}
```

(b) Write the general form of the **while** statement: ***5 marks***

```
initialize loop
while (condition is true) {
  perform purpose of loop
  progress towards end
}
```

(c) The following perl code will compile and run but for any of at least 4 reasons will not generate the desired output. Why?

4 * 5 marks

```
#!/usr/bin/perl
# A short demonstration program.
# This program should write out all the numbers
# from 1 to 20, one number per line.
$control = 20;
$counter = 1;
while ($counter lt 20)
{
  # This next line prints the number
  print "$control\t";
  $counter=$counter+2;
}
```

1	**$counter=$counter+2; # loop jumps by 2 instead of 1**
2	**print "$control\t"; # output variable should be $counter**
3	**print "$control\t"; # \t is for tab – not new line (\n)**
4	**while ($counter lt 20) # lt for comparing text: use <=**
5	**while ($counter lt 20) # *also:* should be <=**
X	**le** *vs* **lt** may seem correct – but either text comparator will fail – so **4** and/or **5** above are correct

Question 3. Total 40 marks.

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

3 x 10 marks

```
#!/usr/bin/perl
$n = 2;
$a = $n * 10;
while ($n <= 8)
{
  $c = $n * 2;
  $n++;
}
$b = $a + $c;
$c = $n * 2;
$t = $b;
$b = $a;
$a = $t;

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

Variable	*Value*
$a	**36**
$b	**20**
$c	**18**

(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%c%c%c%c\n",
35,75,101,101,112,32,73,116,32,83,105,109,112,108,101,35;
```

#Keep It Simple#

Question 4. Total 40 marks.

To convert *miles* to *kilometers* - *divide* by *5* and *multiply* by *8*	k=(m/5)*8
To convert *kilometers* to *miles* - *divide* by *8* and *multiply* by *5*	m=(k/8)*5

Write a perl program to:

1) Present a simple menu to show conversion options.
2) Take all steps to perform the conversion requested.

Include error checking. Indent and comment as appropriate. ***40 marks***

```
#!/usr/bin/perl
# This is a suggested solution only.
# Program to perform distance conversions
# Only 'M' and 'K' are accepted as valid choices
# Use of OR (||) in conditions would deal with lowercase
# entries - for simplicity left out at this time.
print "Program to convert Miles to Kilometers or Kilometers to
Miles.\n";
print "Please choose (M) to convert from Miles\n";
print "or (K) to convert from Kilometers: ";
$choice = <STDIN>;
chomp $choice; # remove EOL character - just in case!
if ($choice eq "M") {
  print "Enter a distance: ";
  $m = <STDIN>;
  if ($m != 0) { # avoid division by zero
    $k = ($m/5)*8; # Use the conversion formula given
  }
  else {
    $k = 0; # deal with zero as input
  }
  print "The distance in Kilometers is $k.\n";
}
elsif ($choice eq "K") {
  print "Enter a distance: ";
  $k = <STDIN>;
  if ($k != 0) { # avoid division by zero
    $m = ($k/8)*5; # Use the conversion formula given
  }
  else {
    $m = 0; # deal with zero as input
  }
  print "The distance in Miles is $m.\n";
}
else { # generate error message if input choice is not valid
  print "\n An incorrect choice was made.\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											