

Question 1. Total 40 marks.

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

6 * 5 marks

```
1:  #!/usr/bin/perl
2:  use strict;
3:  #
4:  my (@sentence, $one, $count, $need)
5:  @sentence=split("-", "each-and-every-part.");
6:  Scount = 0;
7:  foreach $one (@sentence) {
8:      print "$count $one \n"; }
9:      $count++;
10: }
11: elsif ($count > 10) {
12:     print 'lotsofwords';
13: }
```

| | |
|---|---|
| 1 | Line 4, semi-colon |
| 2 | Line 6, S instead of \$ (Scount instead of \$count) |
| 3 | Line 9, should be ++ |
| 4 | Line 10, bracket mis-match |
| 5 | Line 11, unexpected elsif symbol |
| 6 | Line 12, single/double commas don't match |

(b) There is one error in this code snippet. Mark the error and identify it as either a *syntax error* or a *semantic error*. **5 marks**

```
print "Enter a number: ";
$data == <STDIN>; <-semantic error
if ($data > 500) {
    print "Bignum";
}
```

(c) Why is indentation used in computer programs?

5 marks

To aid program readability and to help convey program structure.

Question 2. Total 40 marks.

(a) Write a **while** loop that writes out the even numbers from 50 to 10: **10 marks**

```
$sentinel = 10;
$control = 50;
while ($control >= $sentinel) {
    print "$control ";
    $control=$control-2;
}
```

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

```
if(condition1) {
    action if condition1 is true;
}
elsif(condition2) {
    action if condition2 is true;
}
else {
    action if both condition1 and condition2 are false;
}
```

(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

```
1: #!/usr/bin/perl
2: # Convert sample sentence to all lowercase,
3: #
4: my ($sentence, $result, @letters, $letter);
5: $sentence = "THIS is Some SAMple TeXT.";
6: $result = "";
7: @letters = split ("", $sentence);
8: foreach $letter (@sentence) {
9:     if (($letter gt "A") && ($letter le "Z")) {
10:         $letter = chr(ord($letter)+65);
11:     }
12:     $result = $result.$letter;
13: }
14: print "$result\n";
```

1 *line 7, incorrect variable name*

2 *lines 7, 8, @letters and @sentence mixed up*

3 *line 9, should be ge (greater-than-or-equal-to)*

4 *line 10, should be 32, (ASCII 'a' is 97, ASCII 'A' is 65, 97-65=32)*

Question 3. Total 40 marks.

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

#Keep It Simple#

(b) Indicate the values in each of the variables **\$va**, **\$vb** and **\$vc** after this program finishes:

3 * 10 marks

```
#!/usr/bin/perl
use strict;
my ($num, $va, $vb, $vc);
$num = 0;
$va = $num * 2;
while ($num <= 9) {
    $num=$num+1;
}
$vb = 12;
$vc = $num * 2;

print ("$va, $vb, $vc\n");
```

| Variable | Value |
|-------------|-----------|
| \$va | 0 |
| \$vb | 12 |
| \$vc | 20 |

Question 4. Total 40 marks.

| | |
|---|-----------------|
| To convert from <i>Acres</i> to <i>Hectares</i> | ha=acres/2.4711 |
| To convert from <i>Hectares</i> to <i>Acres</i> | acres=ha*2.4711 |

Write a short perl program to:

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

Include some error checking. Indent and comment as appropriate.

40 marks

Must avoid dividing by 0, correct if elsif else structures, good attempt at indentation, matched bracket and inverted-comma pairs, logic is good, I/O is appropriate.

Senior
College
Clonmel

