Name: _____

PPSN: _____

Quality and Qualifications Ireland

Computer Programming

C20013

8th May 2014

10:00 - 12:00

Duration: 2 hours

solutions

Instructions:

1. Answer questions in the spaces provided
2. Return this exam paper when finished
3. Write your exam number on your answer book

You may not:

1. You may NOT use a calculator

This exam counts for 40% of the module

Computer Programming 2014

CTI Senior College, Clonmel

Question 1. Total 40 marks.

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

6 * 5 marks

```
 1:     #!/usr/bin/perl
 2:     use struct;
 3:     my ($dayOfWeek);
 4:     my
   @days=("Sun",'Mon","Tues","Weds","thurs","Fri","Sat");
 5:     $dayOFWeek=<STDIN>;
 6:     chomp $dayOfWeek
 7:     foreach (0..6) {
 8:       print "$days[$_]";
 9:       if ($_ == $dayOfWeek) {
10:         print " (that's today)";
11:       }
12:       else {}
13:       print "\n"";
14:     }
15:     print "Done\n";
16:     }
```

1	**2: struct => strict**
2	**4: 'Mon" => "Mon" or 'Mon'**
3	**5: $dayOFWeek -> $dayOfWeek**
4	**6: missing ;**
5	**13: remove extra "**
6	**16: remove }**

(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***

```
$lives--;
if ($lives = 0) { <== semantic error
  print "Game Over";
}
```

(c) What does the sigil @ indicate about a variable? ***5 marks***

That the variable is an array (with more than 1 value)

Question 2. Total 40 marks.

(a) Write a **foreach** loop that writes out the numbers from 5 to 15: ***10 marks***

```
foreach (5..15) {
  print $_;
}
```

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

```
if (condition1 is true) {
 do action 1;
}
elsif (condition2 is true) {
  do action 2;
}
...
else { # if no earlier actions are true
  do default action;
}
```

(c) The following perl web-app 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 input value (assumed to be miles) to
   kilometres
 3:     # (divd by 5 mult by 8)
 4:     use CGI;
 5:     my $gci= new CGI;
 6:     my $miles = $cgi->param('miles');
 7:     my $kilometres;
 8:     if ($miles != 0) {
 9:       $kilometres= ($miles/5)**8;
10:     }
11:     else {
12:       $kilometres=0;
13:     }
14:     print <<endbit;
15:     <html><body>
16:     You entered $miles miles which is Skilometres
   kilometres.
17:     </body></html>
18:     endbit
```

1	**Content-type not printed**
2	**5: variable name misspelled**
3	**9: **8 should be ** **
4	**16: Skilometres should be $kilometres**

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%c%c%c\n", 84, 104,
105, 115, 32, 99, 97, 110, 39, 116, 32, 98, 101, 32, 114, 105,
103, 104, 116;
```

This can't be right

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

5 * 6 marks

```
 1:     #!/usr/bin/perl
 2:     use strict;
 3:     my ($count, $a, $top, $bottom, $step);
 4:     $bottom=1;
 5:     my $top = 50;
 6:     $a=$bottom/$bottom;
 7:     $step = $a * 5;
 8:     $bottom--;
 9:     $count=$bottom;
10:     while ($count <= $top) {
11:       $count=$count + $step;
12:     }
13:     $a = $count * 2;
14:
15:     print ("$a, $top, $bottom, $step, $count\n");
```

Variable	Value
$a	**110**
$top	**50**
$bottom	**0**
$step	**5**
$count	**55**

Question 4. Total 40 marks.

To convert *pounds* weight to *kilograms* weight: *multiply* by 0.453592	kg=lb * 0.453592
To convert *kilograms* weight to *pounds* weight: *multiply* by 2.20462	lb=kg * 2.20462

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***

```
 1:   #!/usr/bin/perl
 2:   use strict;
 3:   # Program to convert kilos to pounds, and back
 4:   # sample solution 2014.Q4
 5:   # NOTE: this is only 1 way of doing this
 6:   # init the control var to be sure loop starts
 7:   my $choice=0;
 8:   my ($inputWeight, $convertedWeight, $units);
 9:   # We'll loop until we get a suitable option. This is our errorchecking
10:   while ($choice == 0) {
11:     print "Choose:\n 1)Convert pounds->kilos\n 2)Convert kilos->pounds\n";
12:     $choice=<STDIN>;
13:     # option 1 is lbs-> Kgs
14:     if ($choice == 1) {
15:       print "Please enter pounds: ";
16:       $inputWeight=<STDIN>;
17:       # apply formula as given
18:       $convertedWeight=$inputWeight*0.453592;
19:       $units="Kg";
20:     }
21:     # option 2 is Kgs-> lbs
22:     elsif ($choice == 2) {
23:       print "Please enter kilos: ";
24:       $inputWeight=<STDIN>;
25:       # apply formula as given
26:       $convertedWeight=$inputWeight*2.20462;
27:       $units="lb";
28:     }
29:     # none of the above? that's an error
30:     else {
31:       # when an error is made, let user know
32:       print "Please make a valid choice!";
33:       # prep. the control variable to go around again
34:       $choice=0;
35:     }
36:   }
37:   # I have the weight units added without an extra if structure
38:   print "The converted answer is $convertedWeight $units\n";
39:   # críoch
```