Perl Syntax

For Questions 1-10 write **ONE** line of Perl code. The questions don’t necessarily follow each other. Each question is worth 1 point, except 17 is worth 3. Submit this Word document with your answers.

1. Using quotation marks, make an array named list with the contents of (in order): hello, one, dos, trois, 4, 5, and last

@list = (“hello”, “one”, “dos”, “trois”, 4, 5, “last”);

1. Repeat #1 but without using any quotation marks.

@list = qw(hello one does trois 4 5 last);

1. Set $x to the 3rd element in @list

$list[2] = $x;

1. Add “blah” as the last element in @list

Push (@list, (“blah”));

1. Removes the first element of @list and places it in $first

$first = shift @list

1. Add the contents of $newfirst to be the new first element of @list.

Splice @list, 0, 0, $newfirst;

1. ‘Clear’ @list (meaning it has 0 elements)?

Splice @list, 0, $#list+1;

1. Set $count to be the number of elements in @list (there are at least two ways to do this)

$count = $#list+1;

1. Set @part to be the first (index 0), third through fifth, and last elements from @list.

@part = @list [0, 2..4, $#list];

1. Replace the 3 elements in list starting at index 4 with the contents of @list2 and “help”

Splice @list, 4, 3, @list2, “help”;

1. Write a C-style for loop using that prints each element of @list one on each line.

for ($i=0; $i < $#list+1; $i++){

print "$list[$i]\n";

}

1. Write a foreach loop that prints each element of @list one on each line.

$index = 0;

foreach $list (@list) {

print "@list[$index]\n";

$index++;

}

1. Any loop of your choice that prints –5 –4 –3 … 3 4 5

for ($i=-5; $i < 6; $i++){

print "$i ";

}

1. Translate the following C++ code in at least three ways, one of which can **NOT** use if.

int x;

…

if (x != 5) {

cout x;

} one way second way third way

if ($x != 5){print $x;} if ($x > 5 || $x < 5){print $x;} unless ($x == 5){print $x;}

1. What happens if you write @lits instead of @list (ie. Misspell the variable name)? What can Perl do to help you catch these errors?

It will still compile but it won’t run. Like if you tried to do a print each loop and misspelled list, it would compile but it wouldn’t print anything. Perl will let you put use warnings; at the top of your file so it will print warnings on the command prompt if something isn’t right. This is useful because the first time I didn’t put that and it didn’t tell me anything so I would otherwise not know what was going on unless I went back through all of my code (which would be a pain if it was longer).

1. Assume you have variables $firstName and $lastName. Write **ONE** line of code that uses the variables names to print the sentence (for example):

Hello my name is Gary Underwood, please call me Gary.

print "Hello my name is $firstName $lastName, please call me $firstName.";

1. (3 points) Circle (or highlight) each of the following pairs that are ==, underline them if they are eq. You are comparing $x and $y.
2. $x=“abc”; $y=‘abc’;
3. $x=“ab2”; $y=“zz2”;
4. $x where $x has never been assigned a value (ie it is undef), $y=0
5. $x where $x has never been assigned a value (ie it is undef), $y=“” (ie the empty string)
6. $x=3.0; $y=3;
7. $x=1/3; $y=0;
8. What is the difference between the “ (double quotes) and ‘ (single quotes)? In cases where they produce the same result, which do you think is (or could be) faster and why (assume compiler doesn’t do optimization to automatically use the faster version for you)?

The difference is single quotes tell perl not to interpret what is inside the quotes and it just leaves it exactly like it looks. With double quotes, perl interprets the variable and replaces it with what the variable equals. like if you had $word = “hello” then double quotes would actually print hello and single quotes would print $word.

I think that single quotes would be faster because it doesn’t have to look and see what the variable name is set to and it just prints exactly what was typed. I think the double quotes is better because if you want a variable in a string of words like number 16 in this lab it is probably because it is something that is going to change otherwise you would’ve just put the name in the string.