



## Project 6: First Perl Program

### 1 Overview

This is your first Perl project. It is designed to ease you into Perl and make use of things you learned in earlier assignments. Your task is to write a Perl program that will read a stream of lines from the standard input stream and process the data on those lines as described herein. Each line contains three positive numbers. For each of these input lines, it will determine whether these three numbers are the sides of a triangle or not. If they are not, it will print a message stating that they are not the sides of a triangle. If they are, it will do some triangle classifying and some computation, and it will print an appropriate message about that line. Details are below.

### 2 Task

The input to the program is a sequence of lines from the standard input stream that you may assume have three positive numbers per line, such as

```
12.0    4.5    82
100     65    75.36
1      2      3
```

and so on. Notice from the above examples that there is no fixed space between the numbers - the amount of space may vary - and that the numbers may be whole or have fractional parts. Let us call the three numbers  $a$ ,  $b$ , and  $c$ . For each line it has to decide if the numbers  $a$ ,  $b$ , and  $c$  form the sides of a triangle. If they do not, it prints the message

```
 $a$ ,  $b$ , and  $c$  are not the sides of a triangle.
```

If the numbers are the sides of a triangle, it then does three things:

1. it determines whether the triangle is acute, obtuse, or right;
2. it determines whether the triangle is scalene (no two sides are the same length), isosceles (two sides exactly are the same length), or equilateral (all three sides are equal);
3. it computes the area of the triangle using **Heron's Formula**:

$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{where } s = (a+b+c)/2$$

and it prints a line of the form

```
 $a$ ,  $b$ , and  $c$  are the sides of a  $\langle t1 \rangle$   $\langle t2 \rangle$  triangle with area  $\langle area \rangle$ 
```

where  $\langle t1 \rangle$  is one of the words acute, obtuse, or right,  $\langle t2 \rangle$  is one of the words scalene, isosceles, or equilateral, and  $\langle area \rangle$  is the computed area. For example, for the input line

```
4 5 3
```

the output should be

```
4, 5, and 3 are the sides of a right scalene triangle with area 6.0
```

When there is no more input, the program terminates.



## 2.1 Handling the Input Data

The input comes from the standard input stream. You have learned enough Perl to read the contents of a line from the standard input stream into a variable. For example, the instruction

```
$line = <STDIN>;
```

stores the contents of the current line in the standard input, up to and including the newline character, into the variable `$line`. If the line has three numbers separated by white space, all of those characters are stored into `$line`. For example, if the input line is this:

```
12.0 16 7.2
```

then the variable `$line` would contain the string “12.0 16 7.2”. If you want the values of the three numbers stored into three separate variables, you cannot use this instruction:

```
($x, $y, $z) = <STDIN>;
```

It will store everything into `$x` and leave `$y` and `$z` empty. To overcome this problem, you will need to use a Perl function that we will learn more about soon, called `split`. The following two lines achieve the goal of storing the three numbers into `$x`, `$y`, and `$z`:

```
$line = <STDIN>;  
($x, $y, $z) = split(' ', $line);
```

where there is a single space between the quotes.

Perl has a square root function named `sqrt`. You can get the square root of any expression *expr* by writing `sqrt(expr)`, as in

```
print “The square root of 10 is “,sqrt(10);
```

## 2.2 Naming Your Program

Name your program `trianglesolver.pl`. I should be able to run it by typing that name on the command line.

## 2.3 Error Handling

There is no error checking requirement in this first Perl program. You can assume that every line has three positive numbers.

## Rubric

This homework is graded on a 100 point scale. The program will be graded primarily on its correctness. This means that it does exactly what the assignment states it must do, in detail. Correctness is worth 70% of the grade. Then it is graded on its documentation and design. Comments and appropriate naming of variables are worth 20%; good design another 7%. Naming all files and directories correctly is 3%.



## Submitting the Program

This assignment is due by the end of the day (i.e. 11:59PM, EST) on Thursday, November 8. (I give a grace period of six hours after that, so it is okay to submit it by 6:00 AM of the following day.)

There is a directory in the CSci Department network whose full path name is

```
/data/biocs/b/student.accounts/cs132/hwks/hwk6.
```

You must put it in that directory. To submit your project, you must follow the instructions below exactly! Do not deviate from these instructions.

To be precise:

1. Login using `ssh` to `eniac.cs.hunter.cuny.edu` with your valid username and password, and then `ssh` into any `cslab` host. Do not forget this step. You will not be able to run the `submithwk` command on `eniac`.
2. If you did not do the work on one of the computers in our network, then upload the Perl program file into your home directory. Create a directory named `your-username-hwk6` and put the Perl program into it.
3. Run the command

```
zip -r your-username-hwk6.zip your-username-hwk6
```

This will create the file `your-username-hwk6.zip`. The `zip` command is a special command that compresses the files in the directory and creates a new file that can later be extracted by the `unzip` command. So it will create a “zip file” named `your-username-hwk5.zip` containing your `your-username-hwk5` directory and the three files it contains. For example, I would run

```
zip -r sweiss-hwk5.zip sweiss-hwk6
```

4. Run the command

```
/data/biocs/b/student.accounts/cs132/bin/submithwk 6 your-username-hwk6.zip
```

Do exactly this. Do not mistype it. The command will create a copy of the file `your-username-hwk6.zip` in the directory

```
/data/biocs/b/student.accounts/cs132/hwks/hwk6
```

It will be named `hwk6_username`, where `username` is your username on the network. You will not be able to read this file, nor will anyone else except for me. If you decide to make any changes and resubmit, just do all the steps again and it will replace the old file with the new one. I will be able to unzip the file, extracting whatever files you created. Do not try to put your file into this directory in any other way - you will be unable to do this.

5. **Do not put anything in this directory other than the file `trianglesolver.pl`. You will lose one point for each file that is in this directory that does not belong there.**

Although these instructions may seem complicated, they simplify the way you submit your work and the way I can retrieve it. If you make mistakes, just start over. If things don't seem to work out, post a question on Piazza with the details included.