**1. Difference between the variables in which chomp function work ?**

**Scalar**: It is denoted by $ symbol. Variable can be a number or a string.

**Array:** Denoted by @ symbol prefix. Arrays are indexed by numbers.

The namespace for these types of variables is different. For Example: @add, $add. The scalar variables are in one table of names or namespace and it can hold single specific information at a time and array variables are in another table of names or namespace. Scalar variables can be either a number or a string

**2. Create a function that is only available inside the scope where it is defined ?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | $pvt = Calculation(5,5);    print("Result = $pvt\n");    sub Calculation{    my ($fstVar, $secndVar) = @\_;    my $square = sub{    return($\_[0] \*\* 2);    };    return(&$square($fstVar) + &$square($secndVar));    }; |

**Output:** Result = 50

**5. Which feature of Perl provides code reusability ? Give any example of that feature.**

Inheritance feature of Perl provides code reusability. In inheritance, the child class can use the methods and property of parent class

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | Package Parent;    Sub foo    {    print("Inside A::foo\n");    }    package Child;    @ISA = (Parent);    package main;    Child->foo();    Child->bar(); |

**6. In Perl we can show the warnings using some options in order to reduce or avoid the errors. What are that options?**

-The -w Command-line option: It will display the list if warning messages regarding the code.

– strict pragma: It forces the user to declare all variables before they can be used using the my() function.

– Using the built-in debugger: It allows the user to scroll through the entire program line by line.

**7. Write the program to process a list of numbers.**

The following program would ask the user to enter numbers when executed and the average of the numbers is shown as the output:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | $sum = 0;    $count = 0;    print "Enter number: ";    $num = <>;    chomp($num);    while ($num >= 0)    {    $count++;    $sum += $num;    print "Enter another number: ";    $num = <>;    chomp($num);    }    print "$count numbers were entered\n";    if ($count > 0)    {    print "The average is ",$sum/$count,"\n";    }    exit(0); |

**8. Does Perl have objects? If yes, then does it force you to use objects? If no, then why?**

Yes, Perl has objects and it doesn’t force you to use objects. Many object oriented modules can be used without understanding objects. But if the program is too large then it is efficient for the programmer to make it object oriented.

**9. Can we load binary extension dynamically?**

Yes, we can load binary extension dynamically but your system supports that. If it doesn’t support, then you can statically compile the extension.

**10. Write a program to concatenate the $firststring and $secondstring and result of these strings should be separated by a single space.**

**Syntax:**

$result = $firststring . ” “.$secondstring;

**Program:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | #!/usr/bin/perl    $firststring = "abcd";    $secondstring = "efgh";    $combine = "$firststring $secondstring";    print "$Combine\n"; |

**Output:**

abcd efgh

**11. How do I replace every TAB character in a file with a comma?**

|  |  |
| --- | --- |
| 1 | perl -pi.bak -e 's/\t/,/g' myfile.txt |

**12. In Perl, there are some arguments that are used frequently. What are that arguments and what do they mean?**

-w (argument shows warning)

-d (use for debug)

-c (which compile only not run)

-e (which executes)

We can also use combination of these like:

-wd

**13. How many types of primary data structures in Perl and what do they mean?**

**The scalar:** It can hold one specific piece of information at a time (string, integer, or reference). It starts with dollar $ sign followed by the Perl identifier and Perl identifier can contain alphanumeric and underscores. It is not allowed to start with a digit. Arrays are simply a list of scalar variables.

**Arrays:** Arrays begin with @ sign. Example of array:

|  |  |
| --- | --- |
| 1 | my @arrayvar = (“string a”, “string b “string c”); |

**Associative arrays:** It also frequently called hashes, are the third major data type in Perl after scalars and arrays. Hashes are named as such because they work very similarly to a common data structure that programmers use in other languages–hash tables. However, hashes in Perl are actually a direct language supported data type.

**14. Which functions in Perl allows you to include a module file or a module and what is the difference between them?**

**“use”**

1. The method is used only for the modules (only to include .pm type file)

2. The included objects are verified at the time of compilation.

3. We don’t need to specify the file extension.

4. loads the module at compile time.

**“require”**

1. The method is used for both libraries and modules.

2. The included objects are verified at the run time.

3. We need to specify the file Extension.

4. Loads at run-time.

suppose we have a module file as “Module.pm”

use Module;

**or**

require “Module.pm”;

(will do the same)

**15. How can you define “my” variables scope in Perl and how it is different from “local” variable scope?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | $test = 2.3456;    {    my $test = 3;    print "In block, $test = $test ";    print "In block, $:: test = $:: test ";    }    print "Outside the block, $test = $test ";    print "Outside the block, $:: test = $::test "; |

**Output:**

In block, $test = 3

In block, $::test = 2.3456

Outside the block, $test = 2.3456

Outside the block, $::test = 2.3456

The scope of “my” variable visibility is in the block only but if we declare one variable local then we can access that from the outside of the block also. ‘my’ creates a new variable, ‘local’ temporarily amends the value of a variable.

**16. Which guidelines by Perl modules must be followed?**

Below are guidelines and are not mandatory

The name of the package should always begin with a capital letter.

The entire file name should have the extension “.pm”.

In case no object oriented technique is used the package should be derived from the Exporter class.

Also if no object oriented techniques are used the module should export its functions and variables to the main namespace using the @EXPORT and @EXPOR\_OK arrays (the use directive is used to load the modules).

**17. How the interpreter is used in Perl?**

Every Perl program must be passed through the Perl interpreter in order to execute. The first line in many Perl programs is something like:

|  |  |
| --- | --- |
| 1 | #!/usr/bin/perl |

The interpreter compiles the program internally into a parse tree. Any words, spaces, or marks after a pound symbol will be ignored by the program interpreter. After converting into parse tree, interpreter executes it immediately. Perl is commonly known as an interpreted language, is not strictly true. Since the interpreter actually does convert the program into byte code before executing it, it is sometimes called an interpreter/compiler. Although the compiled form is not stored as a file.

**18. “The methods defined in the parent class will always override the methods defined in the base class”. What does this statement means?**

The above statement is a concept of Polymorphism in Perl. To clarify the statement, let’s take an example:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | [perl]  package X;    sub foo    {    print("Inside X::foo\n");    }    package Z;    @ISA = (X);    sub foo    {    print("Inside Z::foo\n");    }    package main;    Z->foo();  [/perl] |

**This program displays:**

Inside Z::foo

– In the above example, the foo() method defined in class Z class overrides the inheritance from class X. Polymorphism is mainly used to add or extend the functionality of an existing class without reprogramming the whole class.

**19. For a situation in programming, how can you determine that Perl is a suitable?**

If you need faster execution the Perl will provide you that requirement. There a lot of flexibility in programming if you want to develop a web based application. We do not need to buy the license for Perl because it is free. We can use CPAN (Comprehensive Perl Archive Network), which is one of the largest repositories of free code in the world.

**20. Write syntax to add two arrays together in perl?**

|  |  |
| --- | --- |
| 1 | @arrayvar = (@array1,@array2); |

To accomplish the same, we can also use the push function.

**21. How many types of operators are used in the Perl?**

Arithmetic operators

+, – ,\*

Assignment operators:

+= , -+, \*=

Increment/ decrement operators:

++, —

String concatenation:

‘.’ operator

comparison operators:

==, !=, >, < , >=

Logical operators:

&&, ||, !

**22. If you want to empty an array then how would you do that?**

We can empty an array by setting its length to any –ve number, generally -1 and by assigning null list

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | use strict;    use warnings;    my @checkarray;    if (@checkarray)    {    print "Array is not empty";    }    else    {    print "Array is empty";    } |

**23. Where the command line arguments are stored and if you want to read command-line arguments with Perl, how would you do that?**

The command line arguments in Perl are stored in an array @ARGV.

$ARGV[0] (the first argument)

$ARGV[1] (the second argument) and so on.

$#ARGV is the subscript of the last element of the @ARGV array, so the number of arguments on the command line is $#ARGV + 1

**24. Suppose an array contains @arraycontent=(‘ab’, ‘cd’, ‘ef’, ‘gh’). How to print all the contents of the given array?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | @arraycontent=(‘ab’, ‘cd’, ‘ef’, ‘gh’)    foreach (@arraycontent)    {    print "$\_\n";    } |

**25. What is the use of -w, -t and strict in Perl?**

When we use –w, it gives warnings about the possible interpretation errors in the script.

Strict tells Perl to force checks on the definition and usage of variables. This can be invoked using the use strict command. If there are any unsafe or ambiguous commands in the script, this pragma stops the execution of the script instead of just giving warnings.

When used –t, it switches on taint checking. It forces Perl to check the origin of variables where outside variables cannot be used in sub shell executions and system calls

**26. Write a program to download the contents from www.perlinterview.com/answers.php website in Perl.**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | #!/usr/bin/perl    use strict;    use warnings;    use LWP::Simple;    my $siteurl = 'www.perlinterview.com/answers.php';    my $savefile = 'content.kml';    getstore($siteurl, $savefile); |

**27. Which has the highest precedence, List or Terms? Explain?**

Terms have the highest precedence in Perl. Terms include variables, quotes, expressions in parenthesis etc. List operators have the same level of precedence as terms. Specifically, these operators have very strong left word precedence.

**28. List the data types that Perl can handle?**

Scalars ($): It stores a single value.

Arrays (@): It stores a list of scalar values.

Hashes (%): It stores associative arrays which use a key value as index instead of numerical indexes

**29. Write syntax to use grep function?**

|  |  |
| --- | --- |
| 1  2  3 | grep BLOCK LIST    grep (EXPR, LIST) |

**30. What is the use of -n and -p options?**

The -n and -p options are used to wrap scripts inside loops. The -n option makes the Perl execute the script inside the loop. The -p option also used the same loop as -n loop but in addition to it, it uses continue. If both the -n and -p options are used together the -p option is given the preference.

**31. What is the usage of -i and 0s options?**

The -i option is used to modify the files in-place. This implies that Perl will rename the input file automatically and the output file is opened using the original name. If the -i option is used alone then no backup of the file would be created. Instead -i.bak causes the option to create a backup of the file.

**32. Write a program that explains the symbolic table clearly.**

In Perl, the symbol table is a hash that contains the list of all the names defined in a namespace and it contains all the functions and variables. For example:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | sub Symbols    {    my($hashRef) = shift;    my(%sym);    my(@sym);    %sym = %{$hashRef};      @sym = sort(keys(%sym));    foreach (@sym)    {    printf("%-10.10s| %s\n", $\_, $sym{$\_});    }    }    Symbols(\%Foo::);    package Foo;    $bar = 2;    sub baz {    $bar++;    } |

**33. How can you use Perl warnings and what is the importance to use them?**

The Perl warnings are those in which Perl checks the quality of the code that you have produced. Mandatory warnings highlight problems in the lexical analysis stage. Optional warnings highlight cases of possible anomaly.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | use warnings; # it is same as importing "all"    no warnings; # it is same as unimporting "all"    use warnings::register;    if (warnings::enabled()) {    warnings::warn("any warning");    }    if (warnings::enabled("void")) {    warnings::warn("void", "any warning");    } |

**34. Which statement has an initialization, condition check and increment expressions in its body? Write a syntax to use that statement.**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | for ($count = 10; $count >= 1; $count--)    {    print "$count ";    } |

**35. How can you replace the characters from a string and save the number of replacements?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | #!usr/bin/perl    use strict;    use warnings;    my $string="APerlAReplAFunction";    my $counter = ($string =~ tr/A//);    print "There are $counter As in the given string\n";    print $string; |

**36. Remove the duplicate data from @array=(“perl”,”php”,”perl”,”asp”)**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | sub uniqueentr    {    return keys %{{ map { $\_ => 1 } @\_ }};    }    @array = ("perl","php","perl","asp”);    print join(" ", @array), "\n";    print join(" ", uniqueentr(@array)), "\n"; |

**37. How can information be put into hashes?**

When a hash value is referenced, it is not created. It is only created once a value is assigned to it. The contents of a hash have no literal representation. In case the hash is to be filled at once the unwinding of the hash must be done. The unwinding of hash means the key value pairs in hash can be created using a list, they can be converted from that as well. In this conversion process the even numbered items are placed on the right and are known as values. The items placed on the left are odd numbered and are stored as keys. The hash has no defined internal ordering and hence the user should not rely on any particular ordering.

**Example of creating hash:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | %birthdate = ( Ram => "01-01-1985",      Vinod => "22-12-1983",    Sahil => "13-03-1989",    Sony => "11-09-1991"); |

**38. Why Perl aliases are considered to be faster than references?**

In Perl, aliases are considered to be faster than references because they do not require any dereferencing.

**39. How can memory be managed in Perl?**

Whenever a variable is used in Perl, it occupies some memory space. Since the computer has limited memory the user must be careful of the memory being used by the program. For Example:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | use strict;    open(IN,"in");    my @lines = <IN>    close(IN);    open(OUT,">out");    foreach (@lines)    {    print OUT m/([^\s]+)/,"\n";    }    close(OUT); |

On execution of above program, after reading a file it will print the first word of each line into another file. If the files are too large then the system would run out of memory. To avoid this, the file can be divided into sections.

**40. How can you create anonymous subroutines?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | sub BLOCK    sub PROTO BLOCK    sub ATTRS BLOCK    sub PROTO ATTRS BLOCK |

**41. What do you mean by context of a subroutine?**

It is defined as the type of return value that is expected. You can use a single function that returns different values.

**42. List the prefix dereferencer in Perl.**

$-Scalar variables

%-Hash variables

@-arrays

&-subroutines

Type globs-\*myvar stands for @myvar, %myvar.

**43. In CPAN module, name an instance you use.**

In CPAN, the CGI and DBI are very common packages

**44. What are the advantages of c over Perl?**

There are more development tools for C than for PERL. PERL execute slower than C programs. Perl appears to be an interpreted language but the code is complied on the fly. If you don’t want others to use your Perl code you need to hide your code somehow unlike in C. Without additional tools it is impossible to create an executable of a Perl program

**45. “Perl regular expressions match the longest string possible”. What is the name of this match?**

It is called as “greedy match” because Perl regular expressions normally match the longest string possible.

**46. How can you call a subroutine and identify a subroutine?**

‘&myvariable’ is used to call a sub-routine and ‘&’ is used to identify a sub-routine.

**47. What is use of ‘->’ symbol?**

In Perl, ‘->’ symbol is an infix dereference operator. if the right hand side is an array subscript, hash key or a subroutine, then the left hand side must be a reference.

|  |  |
| --- | --- |
| 1  2  3  4  5 | @array = qw/ abcde/; # array    print "n",$array->[0]; # it is wrong    print "n",$array[0]; #it is correct , @array is an array |

**48. Where do we require ‘chomp’ and what does it mean?**

We can eliminate the new line character by using ‘chomp’. It can used in many different scenarios.For example:

|  |  |
| --- | --- |
| 1  2  3  4  5 | excuteScript.pl FstArgu.    $argu = $ARGV[0];    chomp $argu; --> to get rid of the carrige return. |

**49. What does the’$\_’ symbol mea**n?

The ‘$\_’ is a default variable in Perl and $\_ is known as the “default input and pattern matching space

**50. What interface used in PERL to connect to database? How do you connect to database in Perl?**

We can connect to database using DBI module in Perl.

|  |  |
| --- | --- |
| 1  2  3 | use DBI;    my $dbh = DBI->connect(’dbi:Oracle:orcl’, ‘username’, ‘password’,) |

**52) List the operator used in Perl?**

Operators used in Perl are

* String Concatenation à ‘.’
* Comparison Operators à ==, !=, >,< , >=
* Logical Operators à &&, ll , !
* Assignment Operators à + = ,- + , \*=
* Increment and decrement Operators à ++ ,-
* Arithmetic Operators à +, – ,\*

**53) Explain which feature of PERL provides code reusability?**

To provide code re-usability in PERL inheritance feature is used. In Inheritance, the child class can use the methods and property of the parent class.

**54) Mention the difference between die and exit in Perl?**

Die will print a message to the std err before ending the program while Exit will simply end up the program.

**55) In Perl, what is grep function used for?**

To filter the list and return only those elements that match certain criteria Perl grep function is used.

**56) What is the syntax used in Perl grep function?**

The syntax used in Perl is

a) grep BlOCK LIST

b) grep ( EXPR, LIST )

* **BLOCK:** It contains one or more statements delimited by braces, the last statement determines in the block whether the block will be evaluated true or false.
* **EXPR:** It represents any expression that supports $, particularly a regular expression. Against each element of the list, expression is applied, and if the result of the evaluation is true, the current element will be attached to the returned list
* **LIST:** It is a list of elements or an array

**57) Explain what is the scalar data and scalar variables in Perl?**

Scalar in Perl means a single entity like a number or string. So, the Java concept of int, float, double and string equals to perls scalar and the numbers and strings are exchangeable. While scalar variable is used to store scalar data. It uses $ sign and followed by one or more alphanumeric characters or underscore. It is a case sensitive.

**58) What does -> symbol indicates in Perl?**

In Perl, the arrow – > symbol is used to create or access a particular object of a class.

**59) Mention how many ways you can express string in Perl?**

You can express string in Perl in many ways

For instance “this is techvision.”

* qq/this is techvision like double quoted string/
* qq^this is techvision like double quoted string^
* q/this is techvision/
* q&this is techvision&
* q(this is techvision)

**60) Explain USE and REQUIREMENT statements?**

* REQUIRE statement: It is used to import functions with a global scope such that their objects and functions can be accessed directly

Example: Require Module,

Var=module::method(); //method called with the module reference

* USE statements are interpreted and are executed during parsing, while during run time the require statements are executed.

Example: Use Module

Var=method(); //method can be called directly

**61) Explain what is Chop & Chomp function does?**

* **Chop function** eliminates the last character from expr, each element of the list

* **Chomp function** eliminates the last character from an expr or each element of the list if it matches the value of $/. It is considered better than chop as it only removes the character if there is a match.

**62) Mention what is CPAN?**

CPAN means Comprehensive Perl Archive Network, a large collection of Perl software and documentation.

**63) Explain what is Polymorphism in Perl?**

In Perl, Polymorphism means the methods defined in the base class will always over-ride the methods defined in the parent class.

**64) Mention what are the two ways to get private values inside a subroutine or block?**

There are two ways through which private values can be obtained inside a subroutine or block

* **Local Operator:** On global variables only this operator can operate. The value of the private variable is saved on the **Local Operator** and makes the provision to restore them at the end of the block
* **My Operator:**To define or create a new variable this operator can be used. Variable that is created by **My Operator**will always be declared private to block inside which it is defined.

**65) Explain what is STDIN, STDOUT and STDERR?**

* STDIN: The STDIN file handle is used to read from the keyboard
* STDOUT: It is used to write into the screen or another program
* STDERR: It is also used to write into a screen. STDERR is a standard error stream that is used in Perl.

**66) What is the closure in PERL?**

The closure is a block of code that is used to capture the environment where it is defined. It particularly captures any lexical variables that block consists of and uses in an outer space.

**67) Explain what is Perl one liner?**

One liner is one command line programs and can be executed from the command line immediately.

For example,

# run program under the debugger

perl-d my\_file

**68) Explain what is lvalue?**

An lvalue is a scalar value which can be used to store the result of any expression. Usually, it appears at the left-hand side of the expression and represent a data space in memory.

**70) Explain what is the function that is used to identify how many characters are there in a string?**

To tell how many characters are there in a string, **length ()**function is used.

**71) Explain what are prefix dereferencer and list them out?**

Using a particular prefix when you dereference a variable, they are called prefix dereferencer.

* $- Scalar variables
* %-Hash variables
* @-Arrays
* &-Subroutines
* Type globs-\*myvar stands for @myvar, %myvar

**72) Explain what is the function of Return Value?**

The Return Value function returns a reference to an object blessed into CLASSNAME.