

HSST - I

What are the features of Ruby. Write any 5 strengths of Ruby-Rails

Features of Ruby

- * Ruby is an open-source and freely available on the web but it's subjected to license.
- * Ruby is a general purpose uninterpreted programming language.
- * Ruby is a true object oriented programming language
- * Ruby is a server side scripting language
- * It can be used to write CGI script.
- * Ruby can be embedded into HTML

- * Ruby is similar syntax to many programming languages
- * Ruby is much scalable and big programs written in Ruby are easily maintainable
- * Ruby can be used for developing Internet applications

~~Strength of Ruby Rails is~~

~~Ruby are packed with features that make you more productive~~

~~Meta Programming :-~~

Where other frameworks use extensive code generation from scratch, Rail framework uses meta programming techniques to write programs

Ruby is one of the best languages for meta programming & rails uses this capability

Active record

Rails introduces the active record framework which saves objects into the database.

Convention over Configuration

Most web development frameworks for .NET or Java force you to write pages for configuration code.

Scaffolding

Rails automatically creates most of the scaffolding you'll need.

Built in testing

Rails create simple automated tests you can then extend

Rails also provide supporting code called harness and fixtures

Three Environments

Rails give you three default environment development testing and production.

- ?) what are the Ruby gem commands.
- ?) Build :- Builds a gem for gems
- ?) Cert :- Adjust rubygems certification settings
- ?) Chuk :- Checks installed gems

- 4) Clean-up:- Cleans up old version of uninstalled gems in local repository.
- 5) Contents:- Display the contents of the installed gem
- 6) Dependency:- Shows the dependences of installed gems
- 7) Environment:- Display ruby gems in environmental information
- 8) Help :- Provides help on gem command
- 9) Install :- Installs a gem on local repository
- 10) List:- ~~Display all names whose name starts with STRING~~

Ruby Gems Common Command Options

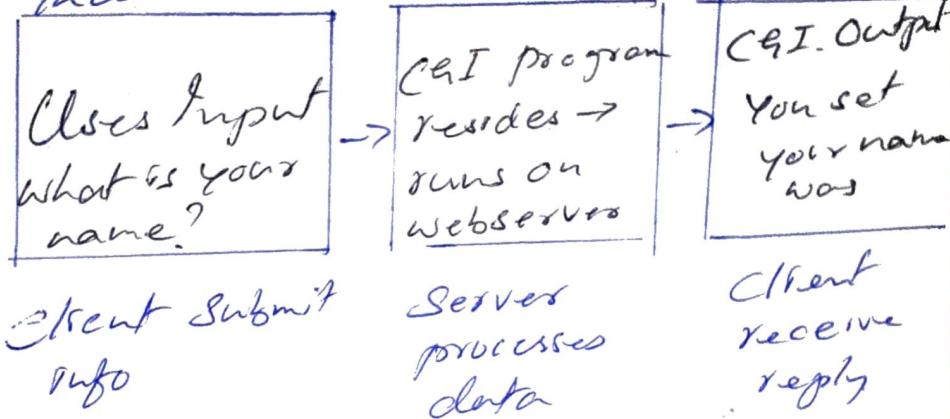
- 1) Source URL :- Uses URL as remote source for gems.
- 2) -P [-no-] http proxy [true] uses HTTP proxy for remote operations
- 3) -h - help gets help on this command
- 4) -- config : -file FILE uses this config file instead of default
- 5) -- backtrace shows stack back trace on errors.
- 6) -- debug turns on ruby debugging

3)

Explain the following

a) CGI Script

- * A CGI Script is any program that runs on a webserver



- * CGI stands for common gateway interface
- * CGI defines a standard in which info may be passed to and from browser to server
- * Any program or script that can process information according to the specification can, in theory, used to code a CGI script.

b) Cookies:-

Generally a server sends some data to the visitors browser in the form of cookies. The browser may accept the cookie.

Expires:- If its blank, the cookie will expire when the visitor quits the browser.

Domain:- The domain of your site

Path:- This may be blank if you retrieve the cookie from any page

Secure:- If the ~~field~~ contains the word secure then the cookie will be only retrieved with a secure server. If this field is blank, no such restriction exists.

Name = Value:- Cookies are set and returned in the form of key and valued pairs.

Ass 2.

Q) How do you define a class in Ruby? Explain with example.

In Ruby, one can create classes and objects easily. Simply write class keyword followed by the name of the class. The first letter of the class name should be in capital letter.

Syntax:-

Class Class-name

End

A class is terminated by end keyword and all the data members are less in between class definition and end keyword.

Eg:- # Class name is fruit

Class ~~Fruit~~

Class variables

`@@ type-of-fruit = 4`

`@@ no-of-fruit = 4`

`end`

2)

Explain the process of memory allocation in Ruby

Memory allocation is the process by which computer programs and resources are assigned with physical or virtual memory space.

You may sometimes need to allocate memory for an extension that won't be used for object storage

Perhaps you have got a grant bitmap for a bloom filter or an image, or a whole bunch of little structures that ruby don't use directly.

In order to work correctly with the garbage collector, you

should use the following memory allocation routines. These routines does a little more work than standard malloc. For instance if Alloc-N determines that it can't allocate desired amount of memory, it will invoke its garbage collector to try its reclaim some space.

If will raise a no mem error if it can't or if the requested amount of memory is invalid.

Memory Allocation

type Alloc-N(C-type, n)

allocate n c-type objects,
where C-type is the ~~literal~~
~~name of the C-type, not a variable~~
of that type. type Alloc(C-type)

Allocates a C-type & casts
the result to a pointer of
that type

Realloc-n (Var, c-type, "n")
reallocates n c-types & assign
the result to var, a pointer
to a C-type, type ACCA-N(
c-type, "n")

Allocate memory for n objects
of c-type on the stack. This
memory will be automatically free'd
when the function that ~~invokes~~
ALLOCATE returns

3)

What is RBS? Explain the key
features of RBS

We defined a new language
called RBS for type signatures
for RBS.

The signatures are written in its files which are different from the ruby code.

You can consider that rbs files are similar to dts files in type script or .h files in c/c++ / objc

The benefit of having different files is, it does not require changing ruby codes to start type checking you can opt in type checking safely without changing any part of the workflow

Key features of RBS

The development of a type system for a dynamically typed language like ruby differs from standard statically typed language

There is lot of ruby code in the world already and a typed system for ruby should support as many of them as possible.

4). How to embedding ruby to other language

When you decide to use an interpreted language such as ruby, you trading raw speed for ease of use.

- * Its for easier to develop a program on a higher level language and you get a working program faster
- * but you sacrifice some of the speed.
- * Thats the simplified view

- * Anyone who spent serious amount of time with working higher level language know that the truth is usually more complex.
- * In many situations the trade off does not really matter
- * But even Ruby 2 codes must admit that there are still situations where its useful to be able to call code written in another language.
- * May be you need a particular part of your program to run extremely fast.
- * Compared to other dynamic language its pretty easy to write C

Extensions in Ruby.

The interfaces you need to understand are easy to use and clearly defined in just a few header files, there are numerous examples available in ruby standard library itself and there are even tools that can help you access all C libraries without writing any C code at all.

Assrt-3.

1) What are the features of Perl? and disadvantages of Perl.

Features of Perl:-

- * Easy to start: Perl is a high level language so it is closer to other popular programming language like C,C++ and thus becomes easy to learn for anyone.
- * Text Processing : As the acronym "Practical Extraction and Reporty Language" suggest that Perl has high text manipulation abilities by which it can generate reports from different text files easily.
- * Contained best features: It contain features of different languages like C, Scala, awk and sh etc which makes the Perl more useful and productive.

* Web and Perl:- Perl can be embedded into web server to increase its processing power and it has the DBI package, which makes web-database integration very easy.

Advantages:-

- ⇒ Perl supports cross platform & it is compatible with most web languages like HTML, XML etc.
- ⇒ It is very efficient in text-manipulation
- ⇒ It is free and open source software which is licensed under Artistic and GNU General Public License (GPL)
- ⇒ It supports more than 25,000 open source modules on CPAN (Comprehensive Perl Archive Network)

- Disadvantages :-
- ⇒ Perl doesn't support portability due to CPAN modules
 - ⇒ Program runs slowly and program needs to be interpreted each time when any changes are made.
 - ⇒ In Perl, the same result can be achieved in several different ways which make the code untidy as well as unreadable
 - ⇒ Usability factor is lower when compared to other language

2)

Write the modern applications of Perl? Explain in detail.

These are mainly three modern applications of Perl

1) Visual Scripting: A collection of visual object is used to construct a graphical interface. This process of constructing a graphical interface

is known as Visual scripting. The properties of visual objects include text on button, background and foreground colors. These properties of objects can be changed by working system is visual basic. It is used to develop new applications. Visual scripting is also used to create enhanced web pages.

2) Scripting Components : In scripting language we use the idea to control the scriptable objects belonging to scripting architecture. Microsoft's visual basic and Excel are the first applications that used the concept of scripting objects. To support all the applications of microsoft the concept of scriptable objects was developed.

3) Web Scripting : Web scripting is classified into three forms. they are processing form, dynamic web pages, dynamically generating HTML

3). What are the scalar expressions of Perl?

Strings and numbers:-

Perl recognizes just two kinds of scalar data: strings and numbers. There is no distinction between integer and real numbers as different types. Perl is a dynamically typed language : the system keeps track of whether a variable contains a numeric value or a string value and the user doesn't have to worry about the difference between

The two kinds of data are done automatically as required by the context in which they are used.

Boolean Values: All programming languages need some way of representing truth values or values of `Bool` is no exception. Some scalar values are either `no`'s or strings, some convention is needed for representing Boolean values and Perl adopts the simple rule that unless zero, "0" and the empty string ("") means false, and anything else means true.

Numeric Constants: Numeric constants can be written in a variety of ways including specific notation, octal and hexadecimal. Although Perl tries to emulate natural human communicating the common practice

of using commas or spaces to break up a large integer constants into meaningful digit groups cannot be used since comma has symbolic significance in Post

String Constants : String constants can be enclosed in single or double quotes. The string is terminated by the first next occurrence of the quote which started it, so a single quoted string can include double quotes and vice versa. The q(quote) and qq (double quote) operators allow you to use any character as a quoting character.

4)

~~Write a short note on~~
a) Arrays b) List c) Hashes

a) Arrays - * An array is an ordered collection of data whose

Comparisons are identified by an ordinal index: This usually the value of an array variable.

- * The name of the variable always start with an @

e.g.: $\text{@rainfall} = (1.2, 0.5, 0.3, 0.1, 0, 0, 0)$

- * A list can occur as an element of another list.

e.g.: $\text{@foo} = (1, 2, 3, \text{"string"})$

$\text{@foobar} = (4, 5, \text{@foo}, 6)$

The foobar result would be $(4, 5, 1, 2, 3, \text{"string"}, 6)$

Q. Lists :-

- * A list is a collection of scalar data items which can be treated as a whole, and has a temporary existence of the runtime stack.
- * It is a collection of variables, constants (number or strings)

to expression which is to be treated as a whole.

- * It is written as a comma separated sequence of values.
eg: "red", "green", "blue".

- * A list often appears in a script enclosed in round brackets.

eg: ("red", "green", "blue")

- * Shorthand notation is acceptable in lists.

eg: ("A", "H", "O", "Z")

- * ~~qwr (the quick brown fox) is a shorthand for ("the", "quick", "brown", "fox")~~

② Hashes:

- * An associative array is one in which each element has two components. a key and a value, the element being "indexed" by its key.

- * Such arrays are usually stored in a hashable to facilitate efficient retrieval and for this reason Perl uses the term hash for an associative array.
- * Names of hashes in Perl starts with an % character: such a name establishes a local context.
- * The index is a string enclosed in braces (curly bracket)
eg: \$somehash{\$_} = 123;
~~\$somehash, "%\$_=0;~~
~~%anotherhash = %somehash;~~

Scripting language Assignment - 04

i) What are the finer points of looping? Explain pack and unpack function

Finer points of looping:

A loop statement allows us to execute a statement or a group of statements multiple times and following is the general form of a loop statement in most of the programming language

```
graph TD; A[A loop statement] --> B[allows us to]; B --> C[execute a statement or a group of statements]; C --> D[multiple times]; D --> E[general form of a loop statement in most of the programming language]
```

The flowchart shows a box labeled "A loop statement" with an arrow pointing down to "allows us to". From "allows us to", an arrow points down to "execute a statement or a group of statements". From there, an arrow points down to "multiple times". Finally, an arrow points down to "general form of a loop statement in most of the programming language".

ii) While loop: Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

iii) Until loop: Repeats a statement or group of statements until a given condition becomes true. It tests the condition before executing the loop body.

- iii) for loop:- Execute a sequence of statement multiple times and abbreviates the code that manages the loop variable.
- iv) for each loop:- The foreach loop iterates over a normal list value and sets the variable VAR to be each element of the list in turn.
- v) do...while loop:- Like a while statement, except that it tests the condition at the end of loop body.
- vi) Nested loops:- You can use one or more loop inside any another while, for or do..while loop.

Pack Function:- The pack function evaluates the expressions in LIST and pack them into a binary structure specified by EXPR. The format is specified using the character.

Each character may be optionally followed by a number, which specifies a repeat count for the type of value being packed, that is nibbles, char or even bits according to the format. A value of `for` as many values remain in `LIST`. Values can be unpacked with the `unpack` function.

Unpack Function:

The `unpack` function unpacks the binary string stored using the format specified in `TEMPLATE`.
Basically reverses the operator of `pack`, returning the list of packed values according to the supported format.

You can also prefix any format with a `/<number>` that you

want a 16-bit checksum of the value of `STRINGS`, instead of the value

- 2) Explain how file handling is done in Perl. write the file handling function in detail.

A file handle is a named internal Perl structure that associates a physical file with a name. All file handles are capable of read/write access, so you can read from and update any file or device associated with file handle. However when you associate a file handle, you can specify the mode in which the file handle is accessed or opened.

Three basic file handles are `STDIN`, `STDOUT` and `STDERR`, which represent standard Input, standard output and standard error devices respectively.

Open Function

The syntax to open file.txt in read only mode. Here less than < sign indicates that file has to be opened in read only mode.

```
open(DATA, "<file.txt");
```

Sysopen Function

The sysopen function is similar to the main open function except that it uses the system open() function using the parameters supplied to it as the parameters for the syntax function

Close Function

To close a filehandle and therefore disassociate the filehandle from the corresponding file, you use the close function. This flushes the filehandle's buffers and closes the system file descriptor.

```
close(DATA)
```

getc Function

The getc function returns a single character from the specified FILEHANDLE or SCALAR of type 'S' specified.

getc FILEHANDLE

getc

read Function

The read function reads a block of information from the buffered filehandle. This function is used to read binary data from the file.

read FILEHANDLE, SCALAR, LENGTH, OFFSET

read FILEHANDLE, SCALAR, LENGTH

Point Function

The point function points the evaluated value of the LIST to FILEHANDLE or to the current output file handle. (STDOUT by default)

print FILEHANDLE LIST

print LIST

print

3) Explain the Perl data structures with examples.

ARRAYS OF ARRAYS

- * In Perl a two dimensional array is constructed by creating an array of references to anonymous arrays.
ex:- @colors = ([35,39,43],[4,5,8],
[32,31,25]);
- * The array constructor converts each comma separated list to an anonymous array in memory and returns a reference and when we write an example like \$colors[0][1] = 39;
\$colors[0] is a reference to an array second subscript represent the element present in that array.
- * A two dimensional array can be dynamically created using push operator to add a reference to an anonymous array to the top level array.
ex:- while (STDIN) {
push @table,[split]
}

Complex data structures

- * Not only an array of arrays can be created but we can create hashes of hashes, arrays of hashes and hashes of arrays.
- * By combining all these possibilities data structures of great complexity can be created

ex: doubly linked list

- * We can make an element of the array a hash containing three fields with keys 'L', 'R' and 'C'
- * The values related to L and R are references to element hashes and the value of C can be anything

Ex:-

We can move forward along the list with

$\$current = \$current \rightarrow \{R\}$;

and backwards with

$\$current = \$current \rightarrow \{L\}$;

Create a new element

$\$new = \{L \rightarrow \text{value}, R \rightarrow \text{undefined}, C \rightarrow \dots\}$

and we can insert new elements
after current element as

$$\$new \rightarrow \{r'\} = \$current \rightarrow \{r'\};$$

$$\$current \{r'\} \Rightarrow \{l'\} = \$new;$$

$$\$current \{r'\} = \$new;$$

$$\$new \rightarrow \{l'\} = \$current;$$

and the current element can be deleted
as

~~$$\$current \rightarrow \{l'\} - \{r'\} = \$current \rightarrow \{r'\};$$~~

~~$$\$current \rightarrow \{r'\} \rightarrow \{l'\} = \$current \rightarrow \{l'\};$$~~

Scripting Languages Assignment - 05

- 1) Write the fundamental features of TCL
- TCL is a string based scripting language and also a procedural language.
- * The language is commonly used for GUI and testing
 - * In TCL by default everything is string
 - * TCL is shell application that reads TCL command from its standard input or from a file and gives desired results.
 - * TCL program should have .tcl extension

- 2) what are the TCL expressions and operators.

~~Expression is constructed from operators and operators. It's evaluated with "expr" command. Operators are evaluated based on precedence and associativity. TCL language has built-in operators~~

<u>Operator Category</u>	<u>Symbol</u>	<u>Procedure/Associativity</u>
Arithmetic Operator	$+-*/\%$	left to right
Relational Operator	$=!=<> <= >=$	left to right
Logical Operator	$\& \& \& \&$	left to right
Bitwise Operator	$+ \& ^ \sim$	left to right
Ternary Operator	$? :$	right to left
Shift Operator	$<< >>$	left to right
String Comparison Operator	$\geq \leq \neq$	left to right
Exponentiation Operator	$**$	left to right
List Operator	$[]$	left to right

3) Explain the following TCC data structures
 a) arrays b) namespaces & procedures

~~Arrays:- An array is a systematic arrangement of a group of elements using indices. The syntax for the conventional array is shown below.~~

~~set ArrayName(Index) value~~

~~ex:- set languages(0) Tel~~

~~set languages(1) "C language"~~

~~put \$ languages(0)~~

~~put \$ languages(1)~~

b) Namespaces-

Namespace is a container for set of identifiers that is used to group variables and procedures. Namespace are available from Tcl Version 8.0.

Before the introduction of the namespace, there was single global scope. Now with namespaces, we have additional partitions of global scope.

c) Procedures-

Procedures are nothing but code blocks with series of commands that provide a specific desirable functionality. It is used to avoid same code being repeated in multiple locations. Procedures are equivalent to the functions used in many programming languages and are made available in TCC with help of proc command

proc procedureName {arguments} {
 body
}

4)

How to trap the errors in TCL and strong manipulation functions

Error handling in TCL is provided with the help of error and catch commands. The syntax for each of these command is shown below.

Error Syntax:-

error message info code.
In the above error command syntax, message is the error message, info is set in the global variable errorinfo and code is set in the global variable errorcode

Catch Syntax:

catch script resultVarName
In the above catch command syntax, script is the code to be executed, resultVarName is variable that holds the error or the result. The catch command returns 0 if there is no error and \ if there is an error
⇒ Strong manipulation function accept argument of type CHAR, NCHAR, VARCHAR, NVARCHAR or UVARCHAR.

5)

Explain the following

- a) Event driven programming in TCL
- b) Nuts and Bolts of socket programming
- c) Security issues
- d) What are the features of TCL

a) Event driven programming.

Event driven is used in long running programs like network servers and graphical user interfaces. This chapter introduces event-driven programming in TCL. TCL provides an easy model in which you register TCL commands and the system then calls those commands when a particular event occurs.

Major classes of events in TCL/Tk

- * GUI events
- + File events
- + Timer events
- + Idle events
- * Virtual events

b) Nuts and Bolts of "Internet programming"

The public Internet is a worldwide computer network that interconnects millions of computing devices throughout the world.

In Internet world, all of these devices are called hosts or end systems. Hosts or end systems are connected together by communication links. These are bidirectionally connected to each other through intermediate switching devices known as packet switches.

End system access the Internet through Internet Service Providers (ISPs) including residential ISPs. The transmission Control Protocol (TCP) and the Internet Protocol (IP) are two of the most important protocols in the Internet.

c) Security Issues:-

The application provides a security model that restricts the execution of the applet. An application might consist

of only a security model with no other functionality and it might be used to run large "applets" that implements major applications such as spreadsheets or word processors.

The security issues associated with applets fall into three major groups: integrity attacks, policy attacks, and denial of service attacks.

Integrity attacks:- A malicious applet may try to modify or delete information in the command in an unauthorized way.

Privacy attacks:- The second form of attack consists of information theft or leakage. A malicious applet may try to steal private info from the host environment and transmit it to conspirators outside the environment.

Denial of Services:- The third form of attack consists of denial of services where the applet attempts to interfere with the normal operation of the host system.

d) Features of TCC

- * It is cross platform with support for Linux, Mac or Unix and Microsoft Windows operation systems.
- * It is an open source.
- * It provides high level of extensibility.
- * It is customizable.
- * It is configurable.
- * It provides a large number of widgets.
- * It can be used with other dynamic languages and not just TCC.
- * GUI looks identical across platforms.

