CHDOLLAR(CH$)

CHDOLLAR PROGRAMMING LANGUAGE

By Jemin Information Technology (C) 2017

About the Author and Preface

This CH$ is Designed by Analzing many Research papers

Using CH$ you can build any Remote WebApplication as fast as could.

I Thank God for this wisdom given to me…

----------Wilmix Jemin J , Jemin Information Technology

This EBOOK is Printed in Asia.

To Make Software Fast like Rabbit movement

and a global redistribution of prosperity

Acknowledgments

We'd like to acknowledge all of the people who played important roles in the creation

of this book. We'd also like to thank all of the developers who've spent time reading this manuscript

and pointing out all of the problems.

Finally, we'd like to extend a sincere thank you to the people who participated in the CHDOLLAR Program. In particular, those who've left feedback in the Author

Online forum have had a strong impact on the quality of the final printed product.

And for providing English translations of the text resources, we'd like to thank

Github and our supporters.

Thanks to all!

---------WILMIX JEMIN J

About this Book

Welcome to CHDollar Remote Web application Programming! If you've picked up this book, we suspect you're a C/C++ or a JAVA Professional.

Working with CH$ who's somehow or other heard about web application like JAVA, DOTNET,etc with client - server concepts.

Perhaps you've worked with the Other Technologies in the past, perhaps you've worked with another Technologies , or perhaps this is your first step into CH$ security with wnosql db. Whichever path has led you here, you're probably looking for a good introduction to the new CH$ Programming Language. This book intends to give you that introduction and much more.

If you've never heard of CHDOLLAR, JAVA7, we cover the basics in enough

depth to keep you in tow. If you know what JAVA7 , CDollar does, but want a deeper understanding of how it does it, we'll provide that too.

Roadmap

Book is focused on CHDollar Programming Language , if you have knowledge or experience about JAVA you can easily focus it.

But Minimum JAVA or C/C++ Technical Knowledge is required to focus on Studying, Designing CHDOLLAR Remote Web application Modules .

CHDOLLAR is a Client – Server based Programming Language focused on

Remote Web application.

The Brief Contents

Code conventions

The following typographical conventions are used throughout the book:

■ Courier typeface is used in all code listings.

■ Courier typeface is used within text for certain code words.

■ Italics are used for emphasis and to introduce new terms.

■ Code annotations are used in place of inline comments in the code. These highlight important concepts or areas of the code.

Code downloads

This will get you the CHDollarSERVER.zip file by downloading it. a couple of CHDollar archive files —as well as some documentation of the source. Instructions on how to install the application are contained in a README file in that download.

CHDOLLAR FOR REMOTE WEB APPLICATION - PART2

==========

UNIT 1: Introduction to CHdollar Programming Language

==========

Definition:

=========

CHDollar Programming Language is a modern technology consists of JAVA OOPS, C OOPS, Behave like CDOLLAR OOPS, GDOLLAR OOPS,used in Remote webapplication , and Advanced OOPs.

It is mainly used as a Remote webapplication.

It is used in case of billing,forms ,normal remotewebapplication ,

CH$-GRAPHICS ,and complex problems.

ABOUT CHDOLLAR Programming Language

It is concenterated in building and designing Advanced Operating systems.

CH$ is invented in GDollar in year 2008.

CH stands for GAMMA and dollar stands for money.

So we called as GAMMA Technology.

Merits of CH$

==========

> It is Good to create any compilers.and Advanced Operating systems and utilities.

> It has simplified code

> it uses Pointers concept.

> CH$ also used with C@ and Arrow# in mobile and cloud computing.

> it is learnable ,easy to use, and let developers to focus only c++ Programming Language

> CH$ also support CHDOLLARAJAX

> it can be used to execute .exe files.

Disadvantages

==========

> It is used to build normal remote webapplication

> absence of framework in CH$.

CHDOLLAR has only 1 pillars which is present in SRGX and jittechsection1

a).CH$(simillar to Advanced OOPS with CUTIL and C++ combination) called as CWEB accepting only .exe and .dll files.

SYNTAX FOR CHDOLLAR (.chdollar) (beautiful syntax)

<CHSAUCER>

@Remote // to load CHDollar packages

<PACK> <namecspacename> // pack will act like namespace

{

<CLASS> <Classname>

{

public void main()

{

<! CHDOLLAR LOGIC !>

}

}

}

</CHSAUCER>

note: This should be saved in filename.CH$

How Chdollar Technology Works? Or chdollar workflow? (Easy Approach than JSTAR)

==========

At first Developers write a CH$ program and compile

Using javac filename.CH$ it creates filename.CH$.exe file for that.now execute the file at url as filename.CH$ when you run remoteserver the .exe output is displayed in browser as a fast manner.

How to run remoteserver in windows?

click on CHDOLLARSERVER located at CHDOLLAR folder.

note : default hostname is localhost or ip address.

Why we need remotewebapplication? what is the difference between webapplication and this?

webapplication webpages will not travel around web but remotewebapplication is seen in internet ,mobiles, and all devices.

==========

UNIT -2: a) CHDollar with CDollar And Advantages over other P.L.

==========

CDOLLARARRAYS

CDollarArrays list1 = new CDollarArrays(string);

to add any collection objects to array use

add (String) functions and to Display those

objects use list1.Display();

Any class that use CDollarArrays you should extends Array in class...

TREEOFARRAY

==========

SYNTAX for TreeofArray:

TreeArray <name> = new TreeArray(String);

<name>.add(elements);

where elements may be string or collections....

LISTOFARRAY

==========

SYNTAX for List of Array:

LArray <name> = new LArray(string);

<name>.add(elements);

The elements may be string or collection elements.

Operators conditions and loops

Operators

+ => ADD

++=> Increment

- => Substract

--=> Substract

* => Mulitply

/ = Division

~ => bitwise unary not operator

NOT (!) => flips true values to false and false values to true.

>>, >>>, and << => IT is used to shift all the bits of a number left

or right

a Specified number of places...

Other Operators

=> != operator

AND => && And operator

OR => || OR operator

?: => value =condition ? value1 : value2 (similar to if then else)

== => compare two values...

= => Assignment operators

Equals => Compare two objects

Relational Operators

> >= => Greater than , Greater than equals.

< <= => Less than , Less than equal

= or != => Equals and not equals

CONDITIONS

IF Syntax:

---

if <condition> statements;

IF then else Syntax:

---

if <condition> statements else statements1

if <condition> statements1 else if condition1 statement2 .... and soon.

SWITCH Statements:

---

switch (expression)

{

case value1 :

statement1;

[break]

.....

case valuen:

statementn;

[break]

---

default:

default_statement;

}

Explanation:

If the expression is equals value1

statement1 will be executed.

if you use break it comes out of the loop

otherwise it continue to execute next statement.

The default value at the end is optional. It can be included if there are other values that can be held in

your variable but that you haven't checked for elsewhere in the switch statement.

THE WHILE LOOP

---

while (<condition>)

{

< Statements block>

}

Note: if the condition is true the block get executed.

otherwise the loop will be continued.

THE DO --- WHILE LOOP

---

do

{

< Statements block>

}

while(<conditon>)

Note: if the condition is true the block get executed.

and it is tested at the end of the loop, but not at the

beginning. The loop will be continued until it satisfies the condition.

biggest reason to be use the do - while loop is that

when you need the body of the loop to be run atleast once.

For Loop

---

for (identifier=value; cond1; iterator operators)

{

< Block statements >

}

For -Each Statement

---

for (variable : collection)

statement;

for eg)

If you add integers (1 to 3) to arraylist

and if you wish to copy and store it in an integer variable

so that you can print the values that is copied from

arraylist.

Then follow this method of for each statements...

for (int a in ar)

{

< Block statements >

}

Output:

---

value=1

value =2

value =3

Continue and Break

Break means it break out of loop

and continue means

it will continue to execute the statements;

for eg)

CHDOLLAR STRUCTURES

CHDollar structure is another user defined data type available in CDollar programming, which allows you to combine data items of different kinds.

It also provide an efficient way of using the same memory location for multi-purpose.

Thus CHDollar Structures is Equivalent = C progamming Structures and union and it uses less memmory capacity than any Programming languages. IT is also used to store collections, object ,etc.

IT is the most important datastructure implemented by wilmix jemin j.

He reduces the demerits of C Programming and Gdollar child is CHDollar. So CHDollar has very beautiful and Advanced Concepts than any Programming Languages.

SYNTAX:

========

<STRUCTURE> <datatype> list = <NEW> <STRUCTURE> <datatype>(object);

list.add(object);

What is Pointers?

Variables that hold memory address are called pointers.

Why we mainly use Pointers?

Pointers reduces the length and complexity of the program,

They increase the execution speed.

It holds the memmory address..

SYNTAX of CHDollar Pointers:

{*}<<DATATYPE>> <pointer-name>
Pointer<<DATATYPE>>(intialvalue);

for eg)

<Str> s ="Life is beautiful";

{*}<<Str>> l Pointer<<Str>>(s);

The given above statement will store the string "Life is beautiful"

in Pointer name l;

BUCKET

Bucket are used to store key,value data, and Generated Random number

where datatype may be string ,object ,etc.

SYNTAX:

Bucket<DATATYPE> list = <NEW>
Bucket<DATATYPE>(<DATATYPEVALUE>);

list.KeyAdd(<DATATYPEVALUE>);

list.add(<DATATYPEVALUE>);

list.RandomAdd();

list.Display(list);

Advantages

Using Bucket you can also Retrieve the values stored n position.

Searching and Insertion is fast than other DTS.

Random Indexing is possible.

eg) If you store a duplicate value such Random key will be different.

It also used to add many values.

EXTEND

Extend class is used in CDollar since to provide multiple inheritence about 100000000 classes . Extends class also list values in methods and constructor values.

Extend means a Bucket contains List of class and it is also

Behave like Bucket. So it is also one of the Advanced concepts in CDollar.

SYNTAX:

EXTEND <<DATATYPE>> list11 = <NEW> EXTEND <<DATATYPE>> (STRING);

list.KeyAdd(<DATATYPEVALUE>);

list.add(<DATATYPEVALUE>);

list.RandomAdd();

list.Display(list);

Advantages:

It is also used to add many values

Indexing is possible

Value can also be list by index and behave like bucket.

It list only the class value and object value.

It is stateless.

PIPE:

PIPE is used to maintain stateful state.

It is used for dataflow in a Program. We can also add the values,

Constructor values of one class and other class and display it.

It also list the values from the Bucket.

SYNTAX:

Pipe <<DATATYPE>> list11 = <NEW> Pipe <<DATATYPE>> (STRING);

list.KeyAdd(<DATATYPEVALUE>);

list.add(<DATATYPEVALUE>);

list.RandomAdd();

list.Display(list);

Why we Prefer CDollar for software Field?

Used in BILLS, forms ,reports,charts, any software project , graphics to web etc.

CHDollar Advantages over JAVA and other Programming Languages

---

A) CHDollar is the combination of C/C++, and Advanced OOPS.

b) CHDollar will only accept the shortest attractive syntax.

c) CHDollar also used for construction of any datastructures.

d) it is easy to focus ,learn and use.

e) It also supports friendly function, pointers , and structures.

f) CHDollar support Virtual memmory and garbage collection.

g) It is efficient, fast and easy to understand, and it is a OOPS Technology.

h) CHDollar is a High level language.

i) CHDollar is highly portable language

j) Using CHDollar you can create any datastructures as libraries and

use it in your Application program.

k) CHDollar language is a structured and object programming language.

l) CHDollar has OOPS concepts like JAVA.

m) CHDollar have the concept of packages,etc.

n) CHDollar have the concept of constructor or destructor and had magic oops concepts.

o) It Support functions with default Arguments

q) It Supports exception handling

r) It have pointer and nodes..

s) CHDollar has simpler oops concepts, which leads to faster development and less mental overhead.

t) CHDollar is almost always explicitly compiled.

u) CHDollar is easy to learn. CHDollar was designed to be easy to use and is therefore easy to write,compile, debug, and learn than other programming languages.

CHDollar is object-oriented. This allows you to create modular programs and reusable code.

v) CHDollar creates .class, .exe or .dll files and it can be used with CHDollar main program.

w) CHDollar will compile and run at same time where other technology can't do

x) CHDollar is mainly used in complex programming , billing the

goods,graphics,etc

y) CHDollar is used with OAKJAVA

z) CHDollar has Advanced OOPS like CDollar , and GDollar.

AA) CHDollar is an interactive Technology.

BB) CHDOLLAR is used as a remote webapplication

FAQS

A) A C /C++ Programmer or any oops developer

can easily study it....

==========

UNIT -3 :CHDOLLAR Program Syntax And its Advanced Concepts .

==========

Syntax:

=======

<CHDollar>

<PACK> <NAMESPACE>

<%

<CLASS> <CLASSNAME>

{

public FLOAT CHDollar-MAIN()

{

<! CHDOLLAR LOGIC!>

%>

?>

BAG

=====

Bag is the extension of LinkedHashmap and it is the fastest

datastructures than Dictionary.

SYNTAX:

=======

Bag object = new Bag();

object .put(key,value);

Functions

getValues(key) => it is used to get the values for a particular key

get(key,loc) => it is used to get the value stored at a loc (indexing

purpose)

boolean containsValue(object Value) => To check the value present in bag or

not.

put(key,value) => it is used to add key and value in Bag

remove(key ,value) => It is used to remove key and value.

TreeList

========

TreeList simillar to Bucket but store items in tree format.

TreeList list = new TreeList ("BUCKETS");

list.KeyAdd(KEY);

list.add(VALUE1);

list.RandomAdd(RANDOMNO);

list.DisplayO(list,0);

MASK

====

It is the extension of Tree Structure and it can store many values

using mask object and we can also retrieve the values stored in mask.

Mask m = new Mask(<DATATYPE>);

m.add(multiple values);

m.getR(Loc); => Get the values stored in right position

m.getL(LOC) => Get the values stored in left position

HEAP:

====

Creates a tree , puts the data into tree in a fairly balanced way and displays

the tree's size and data in a tree by performing an inorder traversal.

Heap hob = new Heap(<datatype>);

hob.add(datum);

hob = new Heap(key,value1,value2);

Bucktist

==========

Bucktist is similalr to Bucket but it is used to addd two values with one

key.

Bucktist l = null;

l= new Bucktist(key,value1,value2);

WICKET

=======

Wicket is used to store multiple values using same object with

4 values per key.

Syntax:

```
Wicket list12;
list12=new Wicket(key,v1,v2,v3,v4);
list12.Display();
list12.Display(list12,location);
```

EXAMPLE -1: BAG

```
<CHDollar>

<PACK> MyP
<%
   <CLASS> Programs
   {
      public FLOAT CHDollar-MAIN()
      {

      Bag  b <NEW> Bag();

b.PUT(1,34);
b.PUT(2,444);
```

```
<PRINTLN>(""+b);




%>

?>
```

EXAMPLE:2 : CHDOLARARRAYS

==========

```
<CHDollar>
<USE>  CUTIL;

<PACK> MyP
{
   <CLASS> Programs
   {

      public FLOAT CHDollar-MAIN()
      {
```

```
<AList>  ar <NEW> <AList> ();
 for  (int  i=0;i<=100000;i++)
 ar.add(i,i);




 <CDOLLARARRAYS> list1 <NEW>  <CDOLLARARRAYS>("ANIMALS ");
       list1.add("1 horse");
 list1.add("2 pig");
 list1.add("3 cow");
 list1.add("4 goat");
 list1.add("5 chicken");

 list1.add("6 ostrich");
 list1.Display();




    %>

?>
```

EXAMPLE-3: CREATE AN BOOTLOADER Using CHDOLLAR

```
 <CHDollar>

<PACK> MYOS
{
   <CLASS> MYOs
   {
public FLOAT CHDollar-MAIN(){


<PRINTLN>("HelloWorld for booting MYOS");



    %>

?>
```

EXAMPE-4: POINTERS

```
<CHDollar>
<PACK> MyP
{
   <CLASS> Programs
   {
      public FLOAT CHDollar-MAIN()
      {


 <Str>  s="dsdds";

 {*} l Pointers (s);

l.add(s);

 for (int i = 0; i NOT= l.size(); i = i + 1)
 {

 <OBJECT> obj=l.GETKEY(i);
 <PRINTLN>(obj);



 }
```

```
    %>

?>
```

Example-6: EXTEND

```
<CHDollar>
<IMPORT>
<PACK> MyP
<%
   <CLASS> Programs
   <%
      public FLOAT CHDollar-MAIN()

      {
```

```
EXTEND list <NEW> EXTEND("BUCKETS");
    list.KeyAdd("1101");
                list.add("jemin");
                list.RandomAdd();
                list.Display(list);
<PRINTLN>(""+list.DisplayO(list,1));

%>
?>
```

EXAMPLE-7: HEAP

```
<CHDollar>
<PACK> MyP
{
    <CLASS> Programs
   {
       public FLOAT CHDollar-MAIN()
      {

Heap root <NEW> Heap("wilmix");
for (int i = 0; i <= 10; i = i + 1)
```

```
{
root.add("item " + i);
}
<PRINTLN>(root.size() );
root.printTree();
    %>

?>
```

Example-8: LArray

```
<CHDollar>
<PACK> MyP
{
   <CLASS> Programs
   {
        public FLOAT CHDollar-MAIN()
       {
 LArray root <NEW> LArray("root");
 root.add("wilmix");
 root.add("jemin");
 root.add("shalom");
```

```
 root.add("1010");
 root.add("101");
 root.add("201");
 root.add("100000000");
 //print the tree's size and contents
 root.printTree();

    %>

?>
```

Example-9 : PIPE

```
<CHDollar>
<PACK> MyP
{
   <CLASS> Programs
   {
public FLOAT CHDollar-MAIN()
       {
Pipe list  <NEW> Pipe("BUCKETS");
     list.KeyAdd("1101");
                 list.add("jemin");
                 list.RandomAdd();
```

```
                list.Display(list);

<PRINTLN>(""+list.DisplayO(list,1));
    %>

?>
```

EXAMPLE-10: TREELIST

```
<CHDollar>
<PACK> MyP
{
   <CLASS> Programs
   {
      public FLOAT CHDollar-MAIN()
      {

TreeList list  <NEW> TreeList ("BUCKETS");
     list.KeyAdd("1101");
                list.add("jemin");
```

```
                list.RandomAdd("1111");
TreeList list2 <NEW> TreeList("BUCKETS");
 list2.KeyAdd("1102");
                 list2.add("rahul");
                 list2.RandomAdd("1112");

<PRINTLN>("DATA="+list.DisplayO(list,0));



<PRINTLN>("DATA="+list2.DisplayO(list2,0));

    %>

?>
```

Example-11 : MASK

```
<CHDollar>
<PACK> My
{
   <CLASS> Programs
   {
        public FLOAT CHDollar-MAIN()
      {
MASK root <NEW> MASK("wilmix");
```

```
for (int i = 0; i NOT= 10; i = i + 1)
 {
 root.add("item " + i);
 }
root <NEW> MASK("root1",1211211,54441);
root  <NEW> MASK("root2",121121,5444);
 root  <NEW> MASK("root5",99121888,"5");

 root  <NEW> MASK("root3",12112,544);
root  <NEW> MASK("root4",1211,54);
root  <NEW> MASK("root51",121,5);
root.printTree();
    %>

?>
```

Example-12 : WICKET

```
<CHDollar>
<PACK> MyPo
{
   <CLASS> Programs
   {
        public FLOAT CHDollar-MAIN()
      {
 Wicket  list12;
```

```
list12 <NEW> Wicket(1000,10002,43433,4343,5555451);
list12 <NEW> Wicket(10001,100021,434331,4343,5555452);
list12 <NEW> Wicket(10002,100022,434332,4343,5555453);
list12 <NEW> Wicket(10003,100023,434333,4343,5555454);
list12 <NEW> Wicket(10004,100024,434334,4343,5555455);
list12 <NEW> Wicket(10005,100025,434335,4343,5555456);


list12.Display(list12);
<PRINTLN>("DATA="+list12.DisplayO(list12,0));

   %>
?>
```

Example-13 : STRUCTURE

```
<CHDollar>
<PACK> MyPoi
{
   <CLASS> Programs
   {
       public FLOAT CHDollar-MAIN()

      {
```

```
 <Str>  s="dsdds";
 {*} l Pointers(s);
l.add(s);
 for (int i = 0; i NOT= l.size(); i = i + 1)
 {

 <OBJECT> obj=l.GETKEY(i);
 <PRINTLN>(obj);


 }

 <STRUCTURE>  list <NEW> <STRUCTURE> (l.GETKEY(0));

 for (int i11 = 0; i11 NOT= list.size(); i11 = i11 + 1)
 {

 <OBJECT>  el=list.ret(i11);
 <PRINTLN>("SNO= "+el);
              }


    %>
?>
```

Example-14 : BUCKETIST

```
<CHDollar>
<PACK> MyP
{
   <CLASS> Programs
   {
        public FLOAT CHDollar-MAIN()
       {

Bucketist  bp   <NEW>  Bucketist("wilmix");
bp   <NEW>  Bucketist(1,222,434);
bp   <NEW>  Bucketist(1,222,434);
bp.Display(bp);
<PRINTLN>("DATA="+bp.DisplayO(bp,1));
    %>
?>
```

==========

UNIT-4: CHDollar RemoteWebApplication Syntax

==========

SYNTAX to execute executable files by remoteserver:

Compile using javac filename.CH$ will create filename.CH$.exe

Rename .exe files as filename.CH$.exe and execute filename.CH$ in browser the .exe output will be

displayed in the browser.

What is the syntax to get and spliting the parameters ?

==========

<AList> <String>
armg=SPLITREQUEST.RESULT(arraylistobject,"arraylist.dsn",howmanyGUI,incrementor);

it is used to get the value from WeBGui like text box ,etc.

Displayhtml.Design1("http://www.hdwallpaperspulse.com/wp-content/uploads/2013/08/08/wallpaper-115621.jpg",250,300); => load this wallpage in jstar webpage

Displayhtml.Design2(); => apply design2

HTML.displayhtml("prog.html"); => apply canvas or html designs

HTML.AnimateHtml("prog2.html"); => apply svg

HTML.displayhtml("r.html"); => apply jquery with JSTAR AJAX

HTML.displayhtml("animate.html"); => perform animation

XMLDATA.TOJSON("rss2.xml"); => convert to json format

ParseXMLDOM.Process(""http://news.google.com/?output=rss"); => convert to rss feed

JTHEMES.WILMIXTHEME1("MOBILE THEME"
,"","weldoen","weldone","","DDSdS","dsdsds","dsdsdsds"); => apply mobile theme

HTML.AnimateHtml("chart2.html"); => perform animation

<PRINTLN>("printstring") =>
It is used to print the string or value to the webpage

HTML.displayhtml("filename.html") => it is used to print the html form contents in the webpage. and it is used in CHDOLLAR-AJAX.

Print.Println(datatype,datatype) for CH$ program.

MY FIRST CH$ PROGRAM

<CHSAUCER>

@Remote

```
<PACK> p
{
  <CLASS> MyArray
  {
    public void main()
    {
      int [] n = new int[15]; /* n is an array of 15 integers */
      int i,j;

ArrayList ar = new ArrayList();

      /* initialize elements of array n */
      for ( i = 0; i < 15; i++ )
      {
        n[ i ] = i + 100;

ar.add(i,n[i]);

      }

      /* output each array element's value */
      for (j = 0; j < 15; j++ )
      {
        Print.Println(""+j,""+ar.get(j).ToString()); // To String to convert object to String.
      }

    }
  }
}

</CHSAUCER>
```

Now compile using javac wil.CH$ what happens?

It produces the .exe file wil.CH$.exe.

Write a CHDOllar Program with GDollar module(use Book dts):

==========

TS.java

=======

```
<CHDollar>
<PACK> TS
<%
   <CLASS> Y
   {
      public FLOAT CHDollar-MAIN()
      {

BOOK u  <NEW> BOOK("Wilmix");

for (int i=0;i<=100;i++)
u.add("col"+i);//add the values to book

for (int i=0;i<=100;i++)
<PRINTLN>(""+u.ret(i));// print the book values

%>

?>
```

Write a Remote webpage program using CH$?

==========

a)tre.CH$

========

```
<CHDollar>
```

```
<PACK> HierachialTree
<%
   <CLASS> roots
   {
      public FLOAT CHDollar-MAIN()
      {

 Bag  tree <NEW> Bag();
  Bag  Member <NEW> Bag();

  //add  a,b,c,d to  class5  section and class6   to  class6 section

Member.PUT("A","CLASS5");
Member.PUT("B","CLASS5");
Member.PUT("C","CLASS5");
Member.PUT("D","CLASS5");
Member.PUT("CLASS6","CLASS6");

//add  s1,s2,s11,s21,s12,s22  slibings.

 Bag  slibings <NEW> Bag();
slibings.PUT("A","s1");
slibings.PUT("A","s2");
slibings.PUT("B","s11");
slibings.PUT("B","s21");
slibings.PUT("C","s12");
slibings.PUT("C","s22");

//add  cricket,read,swin  to  activities..

Bag  activity <NEW> Bag();
activity.PUT("s1","cricket");
activity.PUT("s2","read");
activity.PUT("s21","swim");
activity.PUT("s12","swim");
tree.PUT(1,Member);
tree.PUT(2,slibings);
```

```
tree.PUT(3,activity);


activity.remove("s1","cricket");
//remove cricket activity
tree.remove(3,activity);
//remove activity object
tree.PUT(3,activity);
//again put the activity to 3 index

//so what will be the output?


<PRINTLN>("output="+tree.containsValue(Member));
//this print statement will print output=true

//this print statement will print tree values
  <PRINTLN>("output="+tree);

%>

?>
```

on compilation with CWE EDITOR

will create tre.exe file for futhure use...

and rename it as tre.CH$.exe and execute it in browser.

Write a remote webpage program for that?

b)CH$ remote webpage using BOOK datastructures.

```
<CHDollar>

<PACK> TS
<%
   <CLASS> Y
   {
      public FLOAT CHDollar-MAIN()
      {

BOOK  u   <NEW>  BOOK("Wilmix");

for (int i=0;i<=100;i++)
u.add("col"+i);

for (int i=0;i<=100;i++)
<PRINTLN>(""+u.ret(i));

%>

?>
```

Now follow the same procedure as i said above..

and create a remotewebpage for that.....

c) Create a CH$ using wnosql db

wdba.java
=========

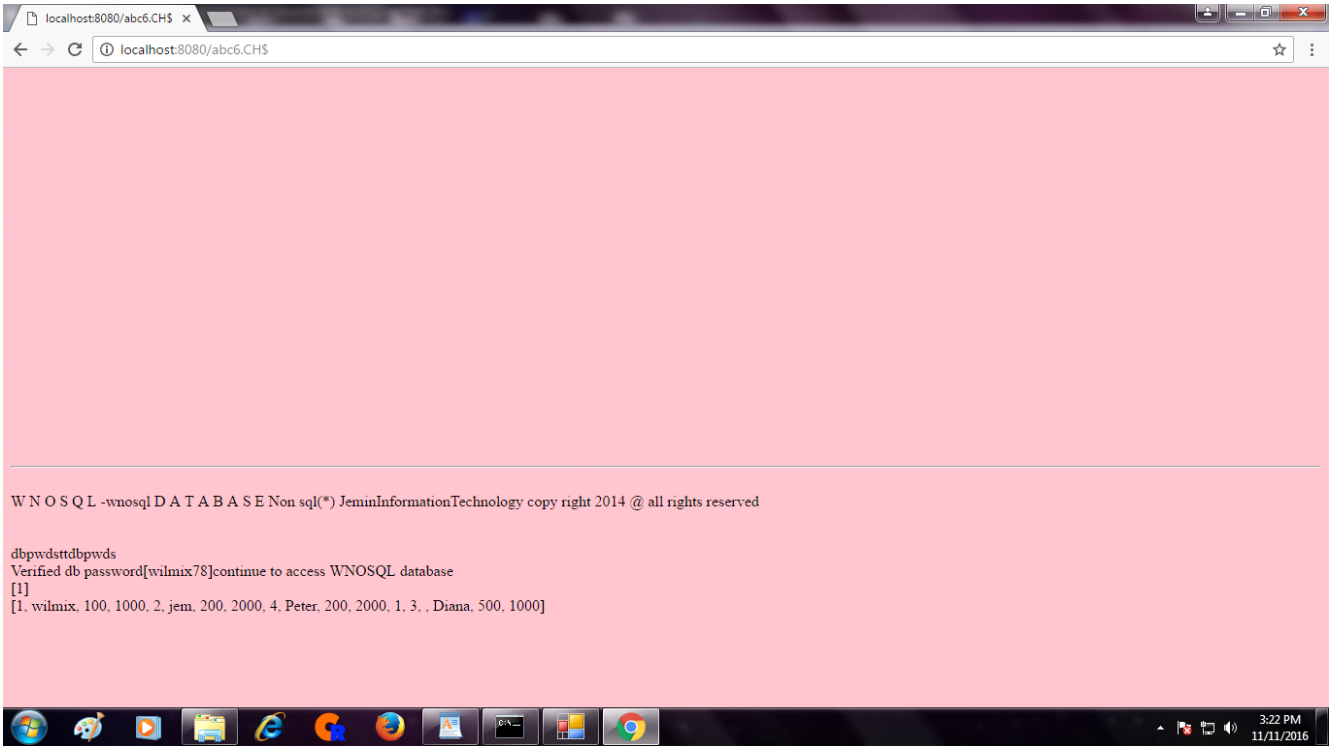
```
<CHDollar>

<USE> CDollar.WDBA;

<USE> WDBA;

<PACK> TSwdba
<%
   <CLASS> Ywdba
   {
      public FLOAT CHDollar-MAIN()
      {



string g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",
"C:\\Prog\\WNOSQ\\WNOSQL\\WNOSQL-cod");
//path where  .wdba  serverer files  is  stored

        string t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);
//supply  username  and  password

  string s11 = "RIGHTJOIN from student 0 to 1 , 1 to 4 ?= emp For X f(x) : {0,1,2,3,4,5,6,7,8,9,10,11} :
{0,1,2,3,4,5,6,7,8,9,10,11} : {0}";

//perform  right  join  between  student  and  employee

 string s1 = "SelectAll from student 0 to 1 , 1 to 4 ?= XXX By X f(x) : {0,1,2}: {3,4,5} :{2,4}";

//select *  from  stodent  from  0  to  1

        <PRINTLN>(""+SQL.WDBAQUERY(  s1,  t));
//print   the   select(*) Query
        <PRINTLN>("" + SQL.WDBAQUERY(s11, t));
//Print  rightjoin  query
```

%>

?>

Compile using CWE EDITOR and create wdba.exe
and rename as wdba.CH$.exe and put it in CH$/bin directory and
rename it as abc6.CH$.exe and execute it in browser.

OUTPUT:



==========

UNIT-5:

=======

CHDollar AJAX ,CHDollar Forms , Bills,etc,Animation and Graphics oops concepts

==========

CHDOLLAR SERVLET

==========

Print.Println("string"," ");

this above statement will print the string on the webpage will

act like chdollar servlet

CHDOLLARAJAX

==========

Print.Println("<form action=BILL.CH$ method=post>","");

if BILL.CH$ is the current CH$ program when you give

method=post and action=CH$ it will act like CH$ AJAX.

CHDOLLAR WXML METHODS

==========

SYNTAX:

=======

WXML.wxml<PARSE>r("filename.wxml", wxmlindex,string+"");

for example)

WXML.wxml<PARSE>r("bill.wxml", 4,s3+"");

=> this statement is used to print bill , Forms,TABLES.

4=> indicates bill format

1=> indicates Forms

2=> indicates TABLES

CHDOLLAR MVC

==========

CHDOLLAR MVC is userdefined it is very easy for developers

to focus like struts.

take 3 class Model(wdba database code) , formview (with set and get methods),

controller .

call controller class result to CH$ main program

ie)

```
public void main()
{



String name= ///parametervalue
//this statements are mandaltory eg)

//to get the value from WeBGui like text box ,etc.


controller.set(name); //set form values.
CHDollar.out.println(controller.get());
//this statement will print the output in webpage
}
```

CH$ GRAPHICS

==========

html1.html

==========

```
<!DOCTYPE html>
<html>
<body>

<p>Image to use:</p>
<img id="scream" src="http://www.en.planettours.bg/02/images/Bodrum/thumbnail.ashx.jpeg" alt="The Scream" width="220" height="277">

<p>Canvas to fill:</p>
<canvas id="myCanvas" width="250" height="300"
style="border:1px solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>

<p><button onclick="myCanvas()">Try it</button></p>

<script>
function myCanvas() {
    var c = document.getElementById("myCanvas");
    var ctx = c.getContext("2d");
    var img = document.getElementById("scream");
    ctx.drawImage(img,10,10);
}
```

```
</script>


</body>
</html>
```

graphics.CH$

==========

```
<CHSAUCER>
<PACK>  p
{
  <CLASS> graphics
  {
    public void main()
    {
HTML.displayhtml("html1.html");
}
}
}
</CHSAUCER>
```

Animate with JQUERY

==========

animate.html

==========

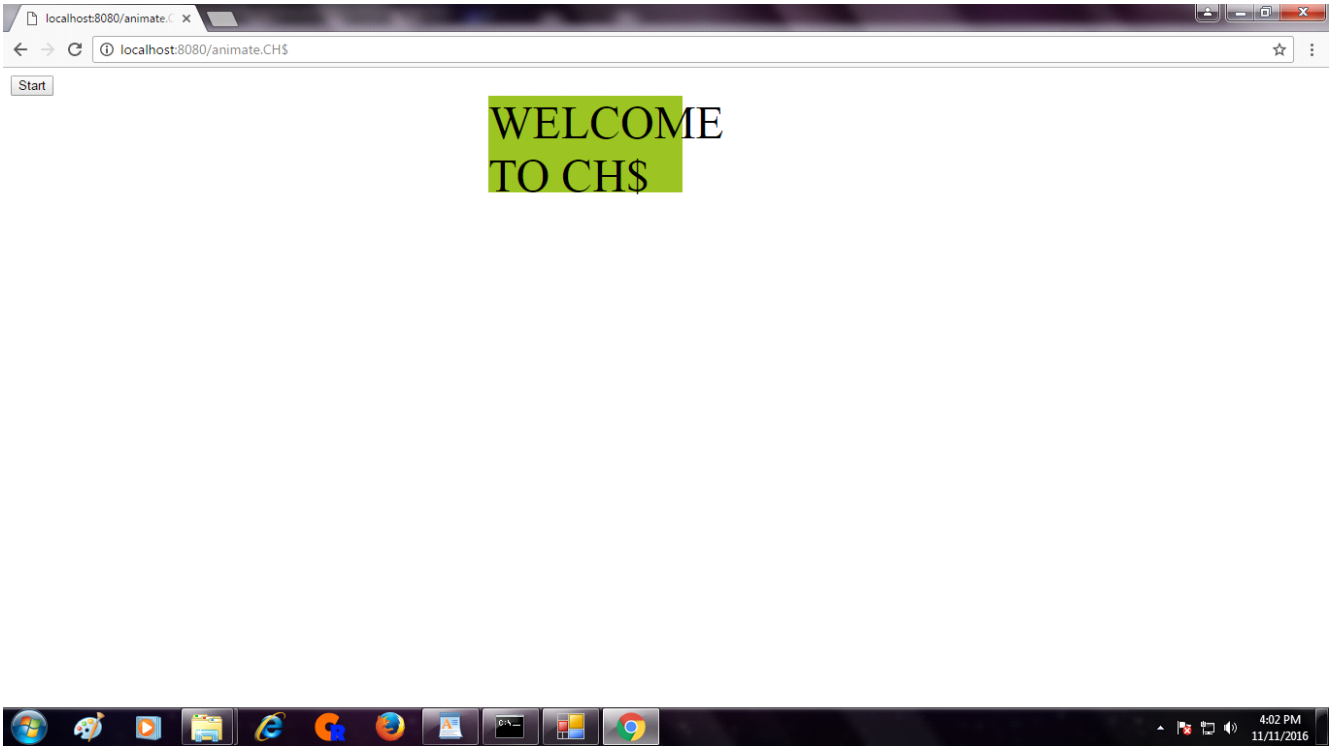
```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        var div = $("div");
        div.animate({left: '500px'}, "slow");
        div.animate({fontSize: '3em'}, "slow");
    });
});
</script>
</head>
<body>
<button>Start </button>
<div style="background:#98bf21;height:100px;width:200px;position:absolute;">WELCOME TO CH$
</div>
</body>
</html>
```

animate.CH$

==========

```
<CHSAUCER>
<PACK>  p
{
  <CLASS> MyArray
  {
    public void main()
    {
HTML.displayhtml("animate.html");
}
}
}
</CHSAUCER>
```

OUTPUT:


localhost:8080/animate.C
localhost:8080/animate.CH$
Start
WELCOME
TO CH$
4:02 PM
11/11/2016

CH$ with ANGULAR JS

==========

ang.html

========

```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
<body>

<div ng-app="">
<p>Kindly type in the text box:</p>
<p>Name : <input type="text" ng-model="name" placeholder="Enter your favourite language"></p>
<h1>Welcome to {{name}} Programming!</h1>
</div>
</body>
</html>
```

angular.CH$

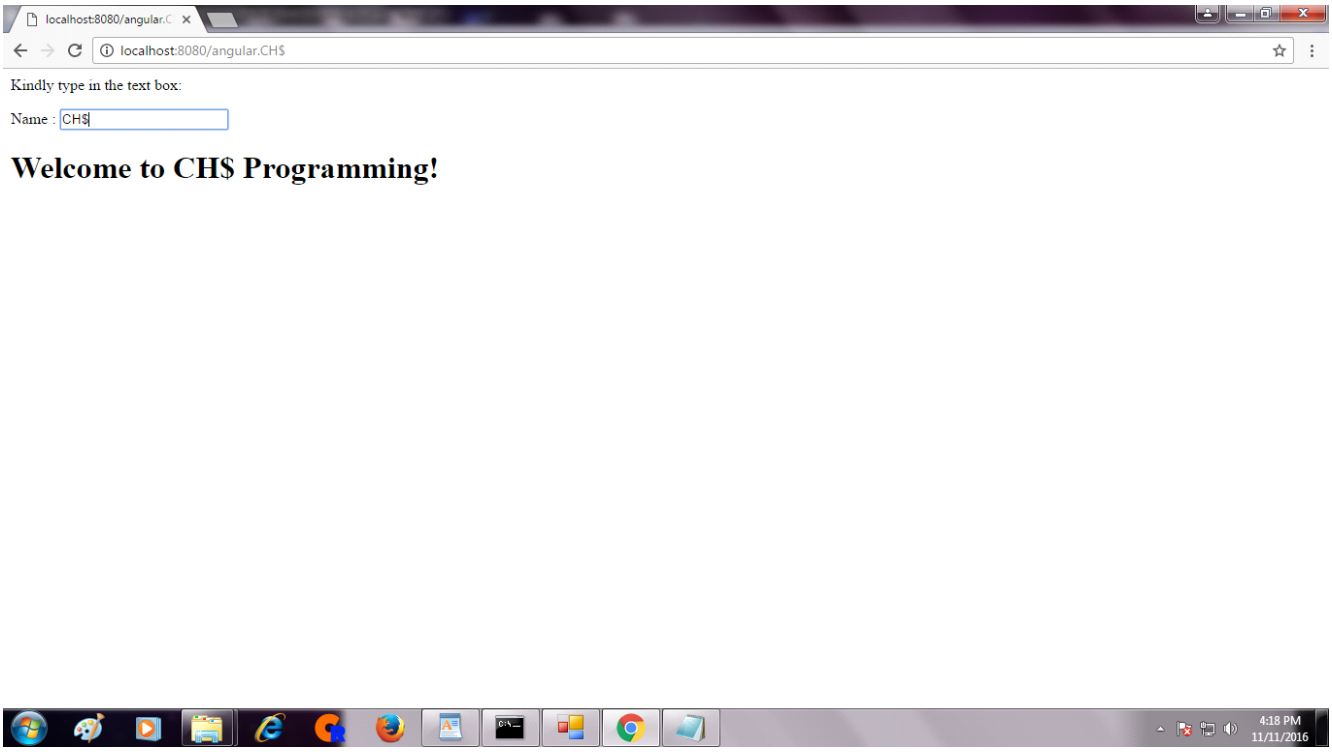
==========

```
<CHSAUCER>
<PACK> p
{
  <CLASS> MyArray
  {
```

```
    public void main()
    {
HTML.displayhtml("ang.html");
}
}
}
</CHSAUCER>
```

OUTPUT

Note: Output is given in next page for user convenience



bootstrap.html

==========

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
  <h2>MY HOMEPAGE</h2>
  <p>MY  HOME  PAGE</p>
  <ul class="breadcrumb">
    <li><a href="#">Home</a></li>
    <li><a href="#">ABOUTME</a></li>
    <li><a href="#">FAQS</a></li>
    <li class="active">Contactme</li>
  </ul>
</div>
</body>
</html>
```

angular.CH$

==========

<CHSAUCER>

<PACK> p

{

<CLASS> MyArray

{

public void main()

{

HTML.displayhtml("bootstrap.html");

}

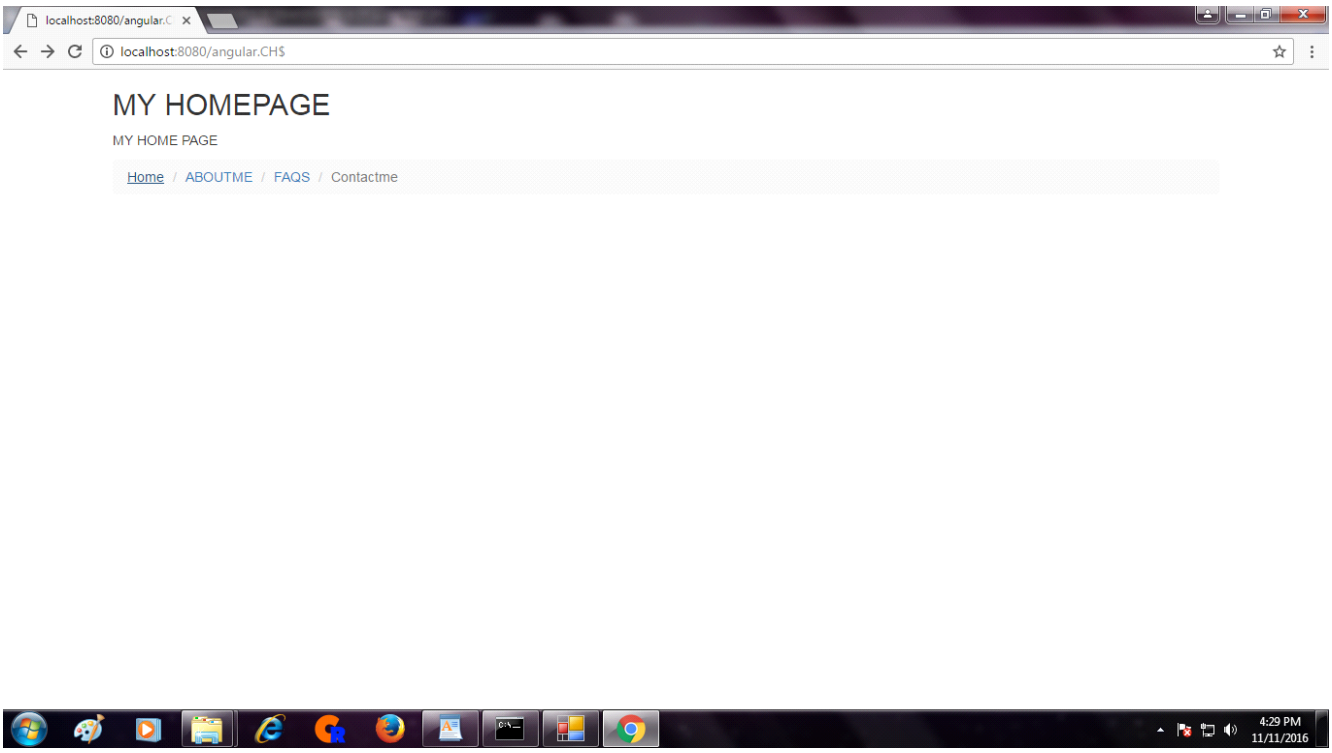
}

}

</CHSAUCER>

OUTPUT:

Note: Output is given in next page for user convenience



Note:

=====

The Necessary files needed for CHDollar is chdollar -javac , chdollar -java and CWEEditor -cdrun to execute .exe files and create .dll files.

CHDollar P.L is designed so that C/C++ and C# Professionals can easily follow it.

index.CH$

```
<CHSAUCER>

<PACK> p
{
```

```
  <CLASS> index
  {
    public void main()
    {

HTML.displayhtml("Register.html");

}
}
}
</CHSAUCER>
```

Register.html

```
<html>
 <head>
  <title>GCLOUDOS  USER  REGISTRATION</title>

 </head>

 <body class="fancy">
<form action="http://localhost:8090/Jquerytest.CH$" method="post" >

<div id="pageContainer">
    <img src="images/banner1.png" alt="GCLOUD OS @  Jemin Information Technology
(C)  ALL  RIGHTS  RESERVED">
    <div id="pageContent">

      <div id="chaptersAccordion">

         <h2><a href="#chapter1">Enter your  System Details</a></h2>
         <div>


<p>Enter  the Name: <input type="text" name="name" size="25" /></p>
<p>Enter your Username : <input type="text" name="uname" size="15"/></p>
<p>Enter  the password : <input type="password" name="password" size="25" /></p>


         </div>


<h2><a href="#chapter2">REGISTER</a></h2>
         <div>
           <input type="submit" name="Click">
<input type="reset" name="Clear">
         </div>
```

```
</form>

</body>
</html>
```

Write a Oak java for Jquerytest as we see this same example in JSTAR.

rename it as Jquerytest.CH$.exe and execute in browser.

Program-2

```
=========

<CHSAUCER>

<PACK> p
{
  <CLASS> index1
  {
    public void main()
    {

HTML.displayhtml("Eregister.html");

}

}
}

</CHSAUCER>


<html>
 <head>
  <title> USER REGISTRATION</title>

 </head>

 <body class="fancy">
<form action="http://localhost:8090/Programs8.CH$" method="post" >

<div id="pageContainer">
```

```
    <div id="pageContent">

      <div id="chaptersAccordion">

          <h2><a href="#chapter1">Enter your  EMP Details</a></h2>
          <div>

<p>Enter  the Name: <input type="text" name="ename" size="25" /></p>
<p>Enter your Username : <input type="text" name="dept" size="15"/></p>
<p>Enter  the password : <input type="password" name="design" size="25" /></p>
          </div>
<h2><a href="#chapter2">REGISTER</a></h2>
          <div>
            <input type="submit" name="Click">
<input type="reset" name="Clear">
          </div>
</form>
</body>
</html>
```

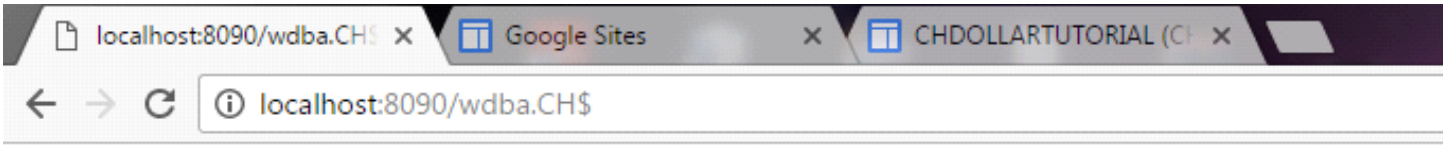
Write a Oak java for Program8 as we see this same example in JSTAR. rename it as Program8.CH$.exe and execute in browser.

==========

EXAMPLE
=========

use wdba.exe and rename it as wdba.CH$.exe

and now excecute in browser as shown below...



0=0, 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
59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86,
111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 1
153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173,
195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215,
237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257,
279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299,
321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341,
363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383,
A, a1, 4111, 10, Next,

CH$ KEYWORDS

abstract boolean break byte

case <CATCH> char class const

continue default do double else

enum <--- final finally float

for goto if --> <USE>

instanceof int interface long native

<NEW> package private protected public

return short Shared strictfp <SUPER>

switch synchronized <IS>

throw throws transient <TRY> void volatile while

<% %>

OTHER KEYWORDS IN CH$

AND -> AND operator
NOT -> NOT operator
-> NOTEQUALS
RUN -> Runnable used in thread
TH-> Thread
<EXE> -> Exception
Friends -> Friend function

OTHER ATTRACTIVE SYMBOLS in CH$

--> => implements

<-- => extends

ARRAYS

=======

ARRAY is to store a value in a location which uses stack dts..

REMOTEWEBAPILCATION-TYPE-2

Remote Type -2 Web application means Remote cloud webapplication

With client and Server.Remote Type -2 Web Application is not visible at internet It is only visible when you give the IP Address correctly.

==========

UNIT-6: ONLINE BILLING,FORMS,REPORTS

==========

ABOUT CHDOLLAR PART2

==========

CHDollar Part2 is focused on Online Billing Software
Consider a scenario in which billing software is
created using java swing and we store the shopping
bill data in oracle or mysql database.
Since Oracle sql is attacked by SQL injection so
data security fails.We already know that WNOSQL database
provides high data security and protect your data from
hackers.

SYNTAX

```
<CDollar>
<IMPORT>
<%
public class <classname> {
  public void  CDOLLAR-Main( )
{
<!  CDollar  Logic  code  !>
}
}
%>
?>
```

Example-1: Write a Electricity bill program

using CDollar-CHDollar-Part2.

```
<CDollar>
<IMPORT>
<%

public class electricitybill {

  public void  CDOLLAR-Main( )
{

HTML.displayhtml("BILL.html");

}
}

%>

?>
```

Write a OAKJAVA7 Model class Program for electricity bill

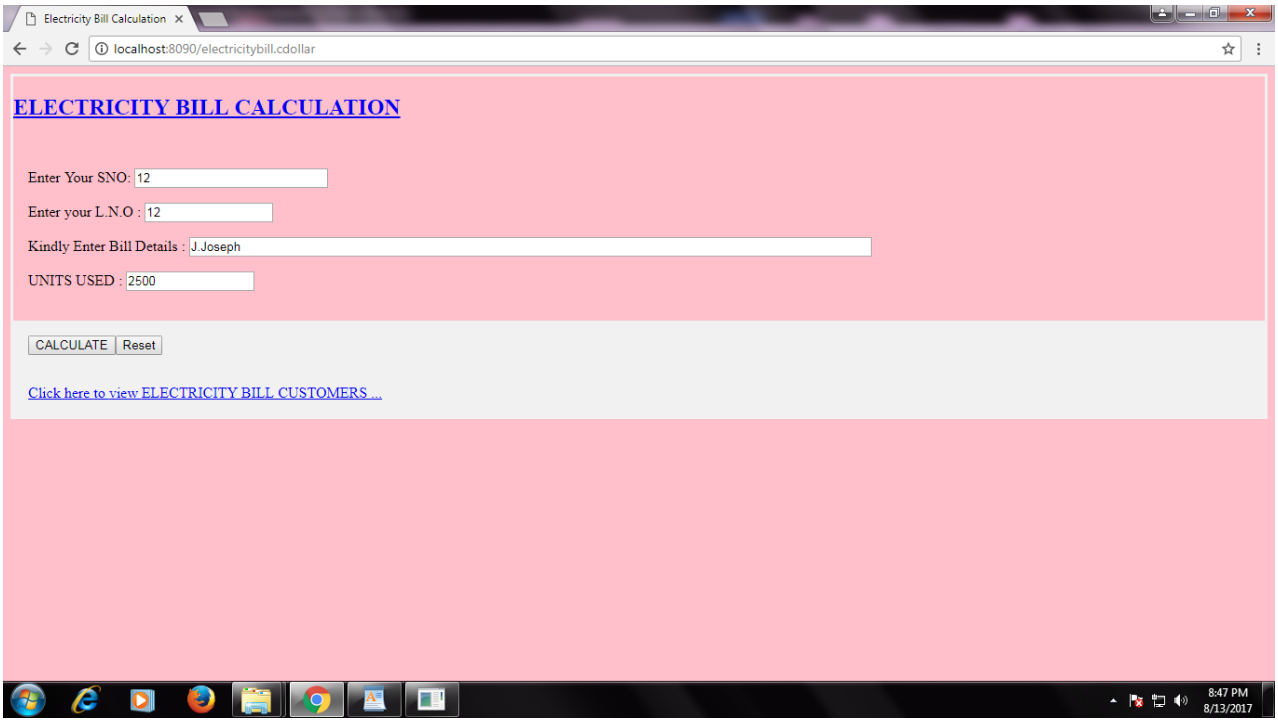
```
<WEB>
<USE> <CDOLLARS>.util;//load  util  packages
<USE> Security; //load  security  packages
<USE> CDollar.WDBA; //  load  Cdollar.wdba packages
<USE> WDBA; //load  wdba  packages
<PACK> Program8
{
   <CLASS> Prog
  {

     static double pressure = 5.5;
```

```
      public static void totalBill(double consumerUnit)
      {
         double Month =( pressure * consumerUnit) * 1;
<PRINTLN>("<td height=200 bgcolor=white>  <font size=4 color=blue>"+ Month+"/-</font></td>");

      }

    public void main()
    {
ArrayList arm1= new ArrayList();
arm1.add("Sno");
arm1.add("Lno");
arm1.add("Billdetails");
arm1.add("Units");
arm1.add("NOT");
<PRINTLN>("<HTML>");
<PRINTLN>("<head> <style>");
<PRINTLN>("table, th, td {");
  <PRINTLN>(" border: 1px solid black; ");
<PRINTLN>("}");
<PRINTLN>("</style>");
<PRINTLN>("</head>");
<PRINTLN>("<BODY bgcolor=pink>");
<PRINTLN>("<form>");
ArrayList  armg= Request.Query(arm1,"electricitybill.cl.dsn",4,1);
 string  s=armg.get(0).ToString();
 <PRINTLN>("<table style='width:100%;' cellpadding=10 cellspacing=5 bgcolor=gold >");
<PRINTLN>("<tr>");
<PRINTLN>("  <p align=center><font size=6 color=blue>TAMILNADU  ELECTRIC  SUPPLY
UNIT</font></p> ");
<PRINTLN>("  <p align=center><font size=3 color=red>ELECTRIC  SUPPLY RECEIPT</font></p>
");
<PRINTLN>(" <p align=left><font size=3 color=blue>Name:</font> </p><p align=right><font size=3
color=blue>SNO:"+s+"</font></p>");
<PRINTLN>("<p align=left><font size=3 color=blue>Electricity No:</font></p>");
<PRINTLN>("<p align=left><font size=3 color=blue>Receipt NO:</font></p><p align=right><font
size=3 color=blue>DAY:</font></p>");
```

```
<PRINTLN>("</tr>");
  <PRINTLN>("<tr>");
<PRINTLN>(" <th><font size=4 color=blue> LNO </font></th>");
<PRINTLN>(" <th><font size=4 color=blue> BILL DETAILS</font></th>");
<PRINTLN>(" <th><font size=4 color=blue> UNITS </font></th>");
<PRINTLN>("<th> <font size=4 color=blue> Amount(Rs)</font></th>");
<PRINTLN>(" </tr>");
<PRINTLN>(" <tr>");
<PRINTLN>(" </tr>");
<PRINTLN>(" <tr>");
<PRINTLN>(" <td height=200  bgcolor=white><font size=4 color=blue>
"+armg.get(1).ToString()+"</font></td>");
<PRINTLN>("<td height=200 bgcolor=white><font size=4
color=blue>"+armg.get(2).ToString().Replace("%40++","@").Replace("%2F","/").Replace("+","
")+"</font></td>");
<PRINTLN>(" <td height=200 bgcolor=white><font size=4 color=blue> "+armg.get(3).ToString()+"
units</font></td>");
double units =Convert.ToDouble(armg.get(3).ToString());
totalBill(units);
<PRINTLN>(" </tr>");
<PRINTLN>(" </tr>");
<PRINTLN>(" <tr>");
<PRINTLN>(" </tr>");
<PRINTLN>("</table>");
  <PRINTLN>("<br>");
  <PRINTLN>("<br>");
  <PRINTLN>("<br>");
  <PRINTLN>("<br>");
  <PRINTLN>("<br>");
  <PRINTLN>("<br>");
<PRINTLN>("<br>");
  <PRINTLN>("<br>");
  <PRINTLN>("<br>");
  <PRINTLN>("<br>");
  <PRINTLN>("<br>");
  <PRINTLN>("<br>");
<PRINTLN>("</form>");
<PRINTLN>("<p align=right><font size=3 color=blue>Electricity  accountant  Signature</font></p>");
```

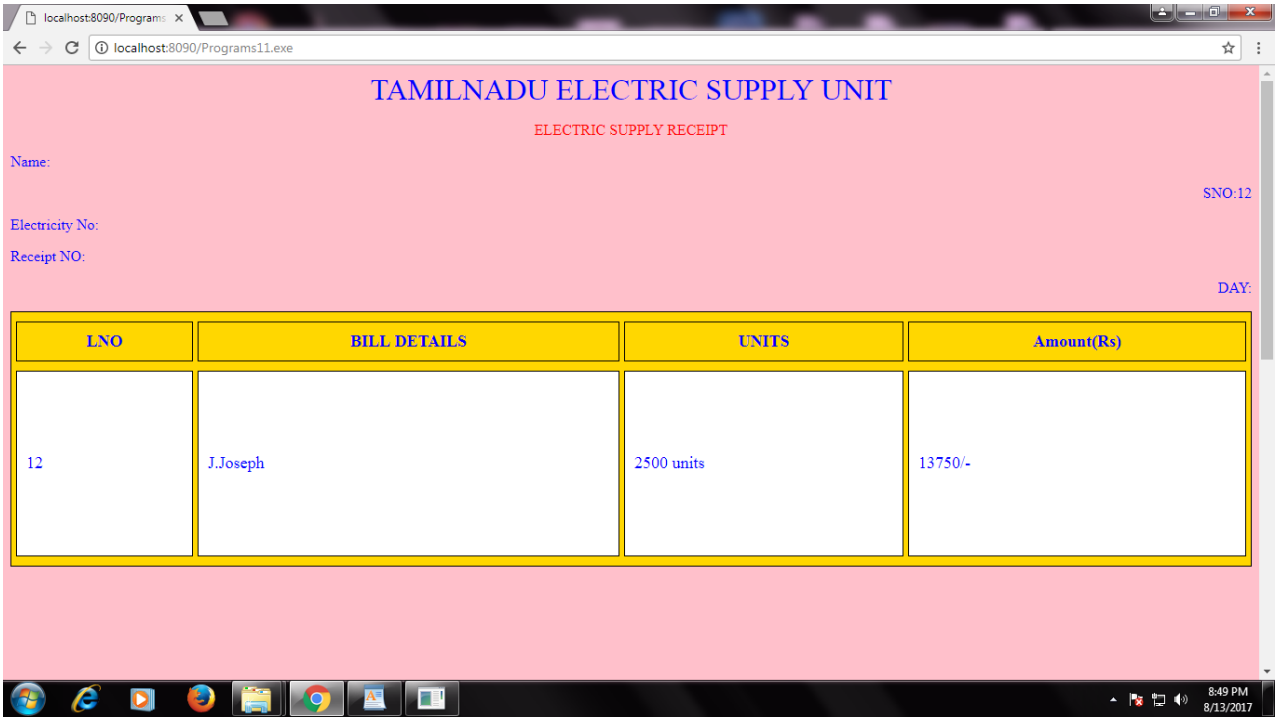
```
<PRINTLN>("</html>");
String g = WDBASQL.WDBASQLS("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL-cod");
        String   t = WDBASQL.WDBASQLS("dbuser", "dbpwds", 1, "wilmix78", "wilmix78", 1, 5, g);
String q = "CREATETABLE from electricitybill 0 to 0 , 1 to 5 ?= 6639 By 6639 f(x) :
{SNO,LNO,BILLDETAILS,UNITS}: {} :{2,4}";
wdbaconn.WDBAQUERY(q);
Char  c= ' ';
ArrayList  datas1=WDBASQL.Query("TABLESIZE()","electricitybill","0",null,19,"","",
null,"",0,"","",c,null,t,1,5);
String t1="";
t1=armg.get(0).ToString()+","+armg.get(1).ToString()+","+armg.get(2).ToString()+","+armg.get(3).ToSt
ring();
String  s12 ="INSERTINTO from electricitybill 0 to "+datas1.size()+" , 1 to 5 ?= A By 1 1 : {0} :
{"+t1+"} : {0}";
  wdbaconn.WDBAQUERY(s12);
<PRINTLN>(" </table>");
<PRINTLN>(" </form>");
<PRINTLN>(" </html>");
      }
   }
}
```

Input:


Electricity Bill Calculation
localhost:8090/electricitybill.cdollar
ELECTRICITY BILL CALCULATION
Enter Your SNO: 12
Enter your L.N.O : 12
Kindly Enter Bill Details : J.Joseph
UNITS USED : 2500
CALCULATE
Reset
Click here to view ELECTRICITY BILL CUSTOMERS ...
8:47 PM
8/13/2017

Output

=======



<u>CHDOLLAR with JDollar(.j$) to list the values from wnosql database.</u>

```
<JDollar>
<USE> <WEB>.util;
<USE> Security;
<USE>  CUTIL;
<PACK> Program5
{
   <CLASS> Prog
  {
     public void Main()
     {
```

```
wdbaconn.JSTARWDBAQUERY("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL-cod");
 wdbaconn.JSTARWDBAUSERQUERY("dbuser", "dbpwds", "wilmix78", "wilmix78");
String  qh2="SELECTRVAL from electricitybill 7 to 24 , 1 to 5 ?= A By 1 1 : {0} : {0} :{0}";
  wdbaconn.WDBAQUERY(qh2);
<TRY>
{
String s=Secure.RetreiveSecure("output.wdba" ,0); //retrieve  the  query  output  from  wdba file
s=s.Replace("[","").Replace("]","");
string []ename = s.Split(' ');
int lengthA = ename.Length;
<PRINTLN>(" <html>");
<PRINTLN>(" <form>");
<PRINTLN>(" <Table  bgcolor=gold>");
  <PRINTLN> ("Electricity customers...");
<PRINTLN> ("<BR>**********");
 ArrayLinearList x =  new ArrayLinearList();
//ArrayLinearList  is  used  to  list
//the reverse  order  of  insertion  data  using
//iterator.

int  c=0;
for  (int  i=7;i<lengthA;i+=1)
{
x.add(c,ename[i].Replace(",","").Replace("%2F","/").Replace("%40","@").Replace("+"," "));

}
<PRINTLN>(" </table>");
<PRINTLN>(" </form>");
<PRINTLN>(" </html>");
      // output using an iterator
      Iterator y = x.iterator();
      while (y.hasNext())
        <PRINTLN>(y.next() + " ");
    Environment.Exit(-1);
}
<CATCH>(<EXE> e){}
```

```
}}}
```

==========

UNIT-7: CH$ JSTAR Part1 and CH$ JSTAR part2

==========

CHDollar Jstar Part1

CHDollar with JSTAR Part1 is focused on biometric system.
Today we use a Biometic system to recognize this person
has done this job or this person has put his
attendance or not.

CHdollar JSTAR Part1 is simillar to JSTAR but
we use it Advanced Projects like biometric system,library management ,etc.

CHDOLLAR-JSTAR

==========

CHDOLLAR-JSTAR is simillar to JSTAR web design.

What is Difference between CHDollar-JSTAR and JSTAR?

==========

JSTAR is focused only on webdesign to be used with php and dotnet. JSTAR webdesign also focus on mobiles.

where on the other hand CHDOllar is focused only on Remote webappilcation -type2 to be used alone.

CHDollar JSTAR webdesign also focus on mobiles.

What are the advantages of CH$-JSTAR over JSTAR?

==========

CH$-JSTAR is used in variety of remote webapplication like biometric,library mgt,etc.

SYNTAX for CHDOLLAR-JSTAR

==========

```
<JSTAR>
<LOGIC>
public   class  <CLASSNAME>
{
public void main( ) throws  <EXE>
{
<!  CHDOLLAR-JSTAR logic  !>
}
}
</LOGIC>
</JSTAR>
```

How can you achieve it ?

You can achieve it using this statement

THUMB IMAGE: <input type="file" name="pic" accept="image/*"> and use this statement in html forms.

TIMEREGISTER.html

```
<html>
<head>
<title>jQuery UI Datepicker - Default functionality</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
  $( function() {
    $( "#datepicker" ).datepicker();
  } );
  </script>
</head>
```

```
<body>
<body bgcolor=pink >
<form name="my" action="http://localhost:8092/TIMEREGISTER1.jstar" method="post" onsubmit="return validateForm();" >
<h1>JIM MENU</h1>
<P align="centre">  <center> <font size="5" color="black" font-family="fantasy" > <h2 id="rabbits-info-heading">SPECIAL GYM</h2></font> <img src="http://wilmix-001-site1.htempurl.com/JIM.jpg" alt="JIM For Body  Building" width="200" height="200"> </center> </p>
<BR>  <BR>
<center><img src="http://wilmix-001-site1.htempurl.com/bodybuilding.jpg" alt="Body  building ..." width="1500" height="700"> <center>
  <h2><a >Enter your  TIME-REGISTRATION Details</a></h2>
        <div>
<p>CustomerNo: <input type="text" name="cid" size="25" /></p>
<p>Enter  the Name: <input type="text" name="name" size="25" /></p>
<p>Enter  the RandomID to  Process: <input type="text" name="rid" size="25" /></p>
<p>THUMB IMAGE: <input type="file" name="pic" accept="image/*"></p>
        </div>
        <div>
 <div ><input type="submit" name="GYMTIMEREGISTER" onClick = "validateForm();">
<input type="reset" name="Clear">
        </div>
</div>
<div ><a href=/JIMMENU.html>Go to GYM MENU</a></div>
    <div><a href=/TIMEREGISTERNEW.html>Click  here  to Enter if  you  are  a  new  customer in timesheet</a></div>

</form>
</body>
</html>
```

TIMEREGISTER1.jstar

```
<JSTAR>
<USE> <J>.net.*;
```

```
<USE>  <J>.util.*;
<USE> <J>.sql.*;
<USE> jxl.Cell;
<USE> jxl.Sheet;
<USE> jxl.Workbook;

<USE> jxl.read.biff.Biff<EXE>;


<USE> jxl.write.Number;
<USE> jxl.write.WritableSheet;
<USE> jxl.write.WritableWorkbook;
<USE> jxl.write.Write<EXE>;

<LOGIC>

public   class  TIMEREGISTER1
{
public void main( ) throws  Biff<EXE> ,<EXE>
{
 <AList><String> arm1= <NEW>  <AList><String>();
arm1.add("cid");
arm1.add("name");
arm1.add("rid");
arm1.add("pic");
arm1.add("!");

<AList><String> armg=SPLITREQUEST.RESULT(arm1,".dsn",4,1);

String  id=armg.get(0).StringConvert();

String EMAIL_REGEX = "^[\\w-_\\.+]*[\\w-_\\.]\\@([\\w]+\\.)+[\\w]+[\\w]$";
     String email1 = id.replace("%40","@");
     Boolean b = email1.matches(EMAIL_REGEX);

String id1="";String ddd1="";String ddd21rid="";
```

```
int  dsdd=0;

String  pwd=armg.get(1).StringConvert();

String s= FingerPrint.call(armg.get(3).StringConvert());
// This  FingerPrint.call  api  is
//  used  for  biometric  system
.....
```

CH$ JSTAR Part2

About CHDollar JSTAR Part2

==========

CHDollar JSTAR Part2 is about CHdollar-Jstar with .j$ we already know that .j$ is used for model class.

Advantages:

CHDollar-jstar with j$ is faster than CHDollar-Jstar.
Used in Projects like Library Management system,etc.

EXAMPLE-1:

```
<JSTAR>
<LOGIC>
public  class  index
{
public void main( )
{
 $HTML.displayhtml("Register.html");
}
}
```

```
</LOGIC>
</JSTAR>
```

Register.html

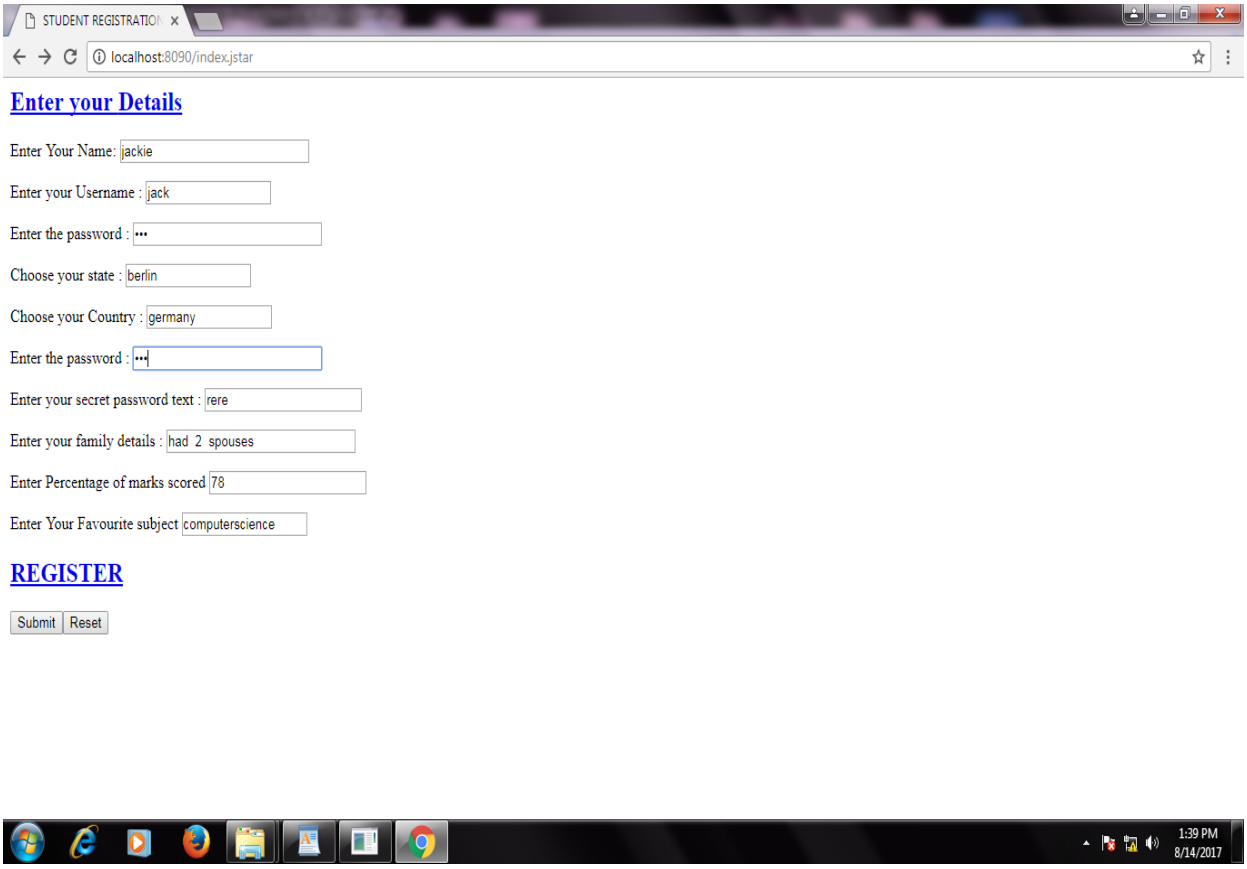
```
<html>
  <head>
    <title>STUDENT  REGISTRATION</title>
  </head>
  <body class="fancy">
<form action="http://localhost:8090/Jquerytest.j$"
method="post" >
<div id="pageContainer">
      <div id="pageContent">
        <div id="chaptersAccordion">
            <h2><a href="#chapter1">Enter your   Details</a></h2>
            <div>
<p>Enter  Your Name: <input type="text" name="name" size="25" /></p>
<p>Enter your Username : <input type="text" name="uname" size="15"/></p>
<p>Enter  the password : <input type="password" name="password" size="25" /></p>
<p>Choose your  state  : <input type="text" name="state" size="15"/></p>
<p>Choose  your  Country  : <input type="text" name="country" size="15"/></p>
<p>Enter  the   password : <input type="password" name="spwd" size="25" /></p>
<p>Enter your secret  password text : <input type="text" name="stext" sixe="15"/></p>
<p>Enter  your family details : <input type="text" name="familydet" size="25" /></p>
<p> Enter  Percentage of  marks  scored <input type="text" name="Indent" sixe="5"/></p>
<p>Enter Your Favourite subject <input type="text" name="CIndent" size="15"/></p>
            </div>
<h2><a href="#chapter2">REGISTER</a></h2><div>
              <input type="submit" name="Click">
<input type="reset" name="Clear">
            </div>
</form>
</body>
</html>
```

Jquerytest1.j$

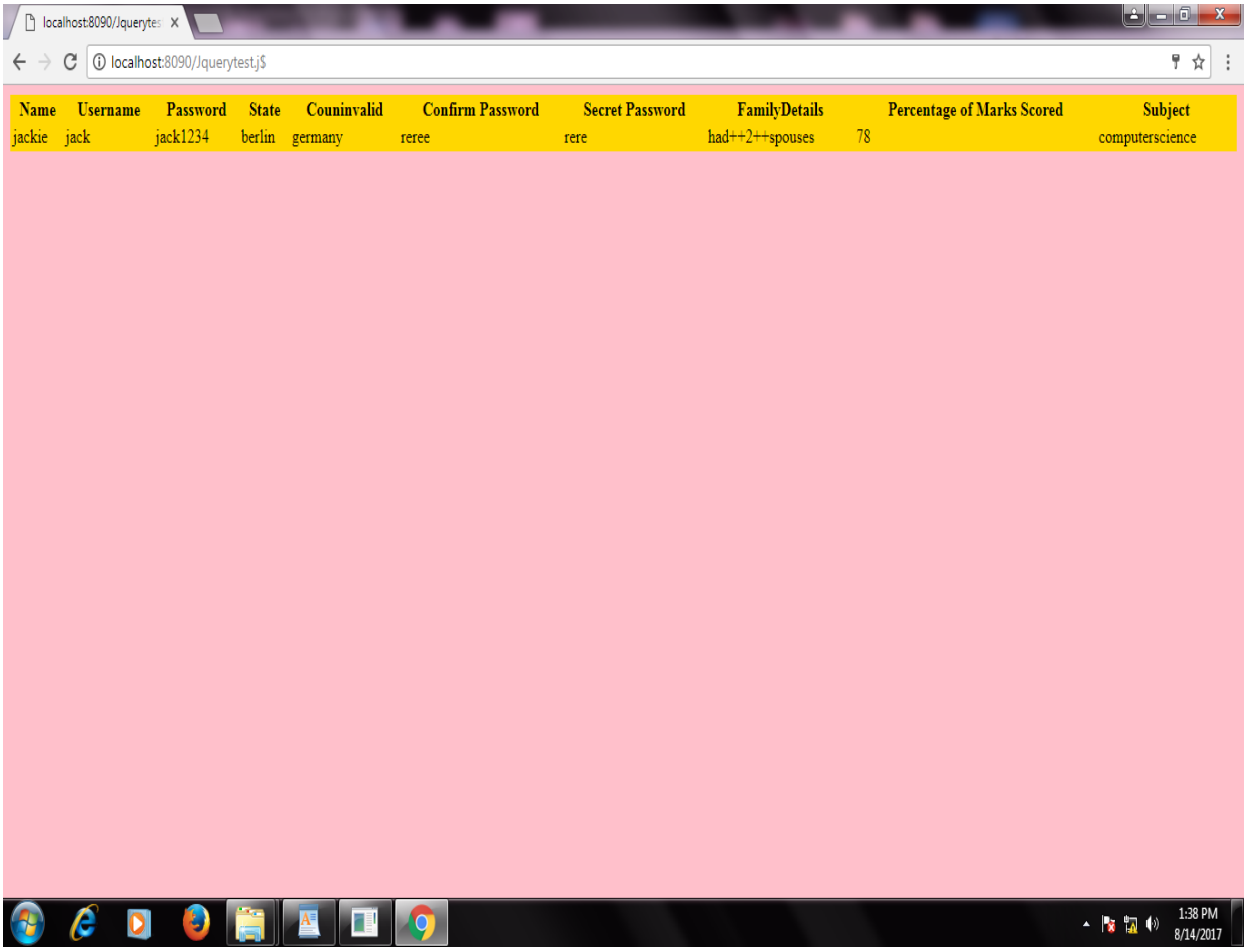
```
===============
<JDollar>
<USE> <WEB>.util;
<PACK> Program5
{
   <CLASS> Prog
  {
     public void Main()
     {
ArrayList arm1= new ArrayList();
//add  parameters
arm1.add("name");
arm1.add("uname");arm1.add("password");
arm1.add("state");
arm1.add("coun<TRY>");
arm1.add("spwd");
arm1.add("stext");
arm1.add("familydet");
arm1.add("Indent");
arm1.add("CIndent");
arm1.add("NOT");
<PRINTLN>("<HTML>");
<PRINTLN>("<BODY bgcolor=pink>");
<PRINTLN>("<form>");
ArrayList  armg= Request.Query(arm1,"index.cl.dsn",10,1);
// get from  10  parameters  values  fom  .dsn
 <PRINTLN>("<table style='width:100%' bgcolor=gold>");
<PRINTLN>("<tr>");
<PRINTLN>("  <th>Name</th>");
<PRINTLN>("  <th>Username</th>");
<PRINTLN>("  <th>Password</th>");
<PRINTLN>("  <th>State</th>");
<PRINTLN>("  <th>Country</th>");
<PRINTLN>("  <th>Confirm Password</th>");
<PRINTLN>("  <th>Secret Password</th>");
```

```
<PRINTLN>("  <th> FamilyDetails </th>");
<PRINTLN>("  <th>Percentage of Marks Scored</th>");
<PRINTLN>("  <th>Subject</th>");
<PRINTLN>(" </tr>");
<PRINTLN>(" <tr>");
for (int i=armg.size()-10;i<armg.size();i++)
{
   <PRINTLN>("<td>"+armg.get(i)+"</td>");
   // print  it  in  table  format
 }
<PRINTLN>(" </tr>");
<PRINTLN>("</table>");
<PRINTLN>("</form>");
<PRINTLN>("</html>");
//  print  statement  for  Jdollar.
}
}
}
```

Input


STUDENT REGISTRATION
localhost:8090/index.jstar
Enter your Details
Enter Your Name: jackie
Enter your Username : jack
Enter the password : •••
Choose your state : berlin
Choose your Country : germany
Enter the password : •••
Enter your secret password text : rere
Enter your family details : had 2 spouses
Enter Percentage of marks scored 78
Enter Your Favourite subject computerscience
REGISTER
Submit
Reset
1:39 PM
8/14/2017

Output:


localhost:8090/Jquerytes
localhost:8090/Jquerytest.j$
Name Username Password State Couninvalid Confirm Password Secret Password FamilyDetails Percentage of Marks Scored Subject
jackie jack jack1234 berlin germany reree rere had++2++spouses 78 computerscience
1:38 PM
8/14/2017

==========

UNIT-8: CH$ ATM Transcation Management

==========

1.1) About CH$ ATM Transcation Management

Today ATM Transcation is done by ATM DEBIT card.
When ATM theives uses duplicate ATM DEBIT CARD with
pin no and password to steal the cash.
If ATM theives cannot steal the cash then he will
break the ATM Box and steal all the money.
So many poor and rich people will loose the money
at any bad time.

1.2) CH$ ATM METHODS

A) ATM User should Don't reveal ur pino and password.

B) Use WNOSQL Security database to store ATM Transcation.

C) CH$ follows Encrpted webpage and when
CH$ is compiled it generates .BS encrypted file.

D) CH$ follows Modern ATM methods.Which is not
reveal for Tutorial Readers.

EXAMPLE-1

index.chdollar

```
<CHDOLLAR>
<CHDollar-Security>
public class  index
{
public void main()
{
HTML.displayhtml("ATMMENU.html");
}
}
?>
```

ATMMENU.html

```
<!Doctype html>
<html>
<body bgcolor=pink >
<div class="centered">
<h1>ATM MENU</h1>
<div class="vertical-menu"  width: 200px>
  <a href="/index.BS" class="active" >ATM MENU</a>
  <a href="/withdraw.chdollar" >WITH DRAW THE  AMOUNT</a>
  <a href="/deposit.chdollar" >DEPOSIT THE  AMOUNT</a>
  <a href="/userlogin.chdollar" >CHECKBALANCE FOR  THE  AMOUNT</a>
  <a href="/register.chdollar" >USER ACCOUNT  REGISTER</a>
  <a href="/LOGOUT1.chdollar">LOGOUT</a>
</div></div>
</p>
</body>
</html>
```

register.chdollar

```
<CHDOLLAR>
<CHDollar-Security>
public class  register
{
public void main()
{
HTML.displayhtml("ATMREGISTER.html");
}
}
?>
```

ATMREGISTER.html

```
<html>
  <head>
    <title>ATM  REGISTRATION</title>
  </head>
  <body class="fancy">
<form action="http://localhost:8090/ATMDETAILS.exe" method="post" >
<div id="pageContainer">
      <div id="pageContent">
        <div id="chaptersAccordion">
<h2><a href="#chapter1">Enter your ATM TRANSCATION  Details</a></h2>
            <div class="container">
            <p>Enter  Your SNO: <input type="text" name="sno" size="25" /></p>
<p>Enter your PINNO : <input type="password" name="pinno" size="15"/></p>
<p>Enter  the RandomId : <input type="password" name="randomid" size="25" /></p>
<p>Enter  the  Amount  to  be  deposited  : <input type="text" name="Amt" size="15"/></p>
          </div><div class="container" style="background-color:#f1f1f1">
            <input type="submit" name="Click">
<input type="reset" name="Clear">
          </div>
 <div class="container" style="background-color:#f1f1f1">
<a href="/index.BS" >GO  BACK </a>
</div>
```

```
</form>
</body>
</html>
```

withdraw.chdollar

```
<CHDOLLAR>
<CHDollar-Security>
public class  withdraw
{
public void main()
{
HTML.displayhtml("WITHDRAW.html");
}
}
?>
```

WITHDRAW.html

```
<html>
  <head>
    <title>ATM  WITHDRAW</title>
  </head>
  <body class="fancy">
<form action="http://localhost:8090/WITHDRAW.exe" method="post" >
<div id="pageContainer">
     <div id="pageContent">
       <div id="chaptersAccordion">
          <h2><a href="#chapter1">Enter your ATM TRANSCATION  Details  for  Withdrawing  the amount</a></h2>
           <div class="container">
<p>Enter the  IDNO : <input type="text" name="idno2" size="15"/></p>
<p>Enter the  AMOUNT  to  be Withdrawn : <input type="text" name="amt2" size="15"/></p>
          </div>
 <div class="container" style="background-color:#f1f1f1">
            <input type="submit" name="Click" value="WITHDRAWAMOUNT">
```

```
<input type="reset" name="Clear">
        </div>
 <div class="container" style="background-color:#f1f1f1">
<a href="/index.BS" >GO  BACK </a>
</div>
</form>
</body>
</html>
```

userlogin.chdollar

```
<CHDOLLAR>
<CHDollar-Security>
public class userlogin
{
public void main()
{
HTML.displayhtml("USERLOGIN.html");
}
}
?>
```

deposit.chdollar

```
<CHDOLLAR>
<CHDollar-Security>
public class  deposit
{
public void main()
{
HTML.displayhtml("DEPOSIT.html");
}
}
?>
```

DEPOSIT.html

```
<html>
 <head>
  <title>ATM  DEPOSIT</title>
 </head>
 <body class="fancy">
<form action="http://localhost:8090/DEPOSIT.exe" method="post" >
<div id="pageContainer">
    <div id="pageContent">
      <div id="chaptersAccordion">
         <h2><a href="#chapter1">Enter your ATM TRANSCATION  Details  for  depositing  the amount</a></h2>
          <div class="container">
<p>Enter the  IDNO : <input type="text" name="idno" size="15"/></p>
<p>Enter the  AMOUNT  to  be deposited : <input type="text" name="amt" size="15"/></p>
         </div>
       <div class="container" style="background-color:#f1f1f1">
            <input type="submit" name="Click" value="DEPOSIT">
<input type="reset" name="Clear">
         </div>
 <div class="container" style="background-color:#f1f1f1">
<a href="/index.BS" >GO  BACK </a>
</div>
</form>
</body>
</html>
```

USERLOGIN.html

```
<html>
 <head>
  <title>ATM  REGISTRATION</title>
 </head>
 <body class="fancy">
<form action="http://localhost:8090/list.exe" method="post" >
<div id="pageContainer">
    <div id="pageContent">
      <div id="chaptersAccordion">
```

```
        <h2><a href="#chapter1">Enter your ATM TRANSCATION  Details</a></h2>
         <div class="container">
<p>Enter your PINNO : <input type="password" name="pinno" size="15"/></p>
        </div>
 <div class="container" style="background-color:#f1f1f1">
          <input type="submit" name="Click" value="Search">
<input type="reset" name="Clear">
        </div>
 <div class="container" style="background-color:#f1f1f1">
<a href="/index.BS" >GO  BACK </a>
</div>
</form>
</body>
</html>
```

List the ATM Customer Details by pino

list.j$

```
<JDollar>
<USE> <WEB>.util;
<USE> Security;

<PACK> Program5
{
   <CLASS> Prog
  {
     public void Main()
     {
ArrayList arm1= new ArrayList();
arm1.add("pinno");
arm1.add("NOT");
<PRINTLN>("<HTML>");
<PRINTLN>("<BODY bgcolor=pink>");
<PRINTLN>("<form>");
 <PRINTLN>(" <BR><a href=/index.BS class=active >Click here ATM MENU</a><BR>");
```

```
ArrayList armg= Request.Query(arm1,"userlogin.BS.dsn",1,1);
/get the parameters form .dsn file and store at armg
wdbaconn.JSTARWDBAQUERY("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQL\\WNOSQL-cod");
 wdbaconn.JSTARWDBAUSERQUERY("dbuser", "dbpwds", "wilmix78", "wilmix78");
String qh2="SELECTRVAL from atmtranscation 5 to 16 , 1 to 5 ?= 1 By 1 1 : {0} : {0} :{0}";
  wdbaconn.WDBAQUERY(qh2);
<TRY>
{
String s=Secure.RetreiveSecure("output.wdba" ,0);
//retrieve the query output from wdba file
s=s.Replace("[","").Replace("]","");
string []ename = s.Split(' ');
int lengthA = ename.Length;
<PRINTLN>(" <html>");
<PRINTLN>(" <form>");
<PRINTLN>(" <Table bgcolor=gold>");
  <PRINTLN> ("ATM customers...");

<PRINTLN> ("<BR>**********");

string[] add = new string[lengthA];
 int n=0;
string d1=armg.get(0).ToString();
for (int i=0;i<lengthA;i+=1)
{
 <PRINTLN>(" <td>");
add[i]=ename[i].Replace("%2F","/").Replace("%40","@").Replace("+"," ").Replace(","," ").ToString();
if (d1==add[i])
{
n=1;
add[i+1]=ename[i+1].Replace("%2F","/").Replace("%40","@").Replace("+"," ").Replace(","
").ToString();
add[i+2]=ename[i+2].Replace("%2F","/").Replace("%40","@").Replace("+"," ").Replace(","
").ToString();
add[i+3]=ename[i+3].Replace("%2F","/").Replace("%40","@").Replace("+"," ").Replace(","
").ToString();
break;
```

```
}
            <PRINTLN>(" </td>");
}
<PRINTLN>("<BR>");
int i1=0;
for  (  i1=0;i1<=add.Length;i1+=1)
{
<PRINTLN>("<td> "+add[i1]+"</td>");
 //print the  table  values
 }
<PRINTLN>(" </table>");
<PRINTLN>(" </form>");
<PRINTLN>(" </html>");
   Environment.Exit(-1);
}
<CATCH>(<EXE> e)
{
}
}
}
}
```

Jdollar_program_to_deposit_the_amount_in atm

```
<JDollar>
<USE> <WEB>.util;
<USE> Security;
<PACK> Program5
{
```

```
   <CLASS> Prog
  {
public static ArrayList tos= new ArrayList();
     public void Main()
     {
ArrayList arm1= new ArrayList();
arm1.add("idno");
arm1.add("amt");
arm1.add("NOT");
<PRINTLN>("<HTML>");
<PRINTLN>("<BODY bgcolor=pink>");
<PRINTLN>("<form>");
ArrayList  armg= Request.Query(arm1,"deposit.BS.dsn",2,1);
wdbaconn.JSTARWDBAQUERY("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL-cod");
 wdbaconn.JSTARWDBAUSERQUERY("dbuser", "dbpwds", "wilmix78", "wilmix78");
String  qh2="SELECTRVAL from atmtranscation 5 to 10 , 1 to 5 ?= 1 By 1 1 : {0} : {0} :{0}";
  wdbaconn.WDBAQUERY(qh2);
 int  n=0;int  xx=0;int  amts=0;int totalamt=0;
<TRY>
{
String s=Secure.RetreiveSecure("output.wdba" ,0);
//retrieve  the  query  output   from   wdba file
s=s.Replace("[","").Replace("]","");
string []ename = s.Split(' ');
 int lengthA = ename.Length;
<PRINTLN>(" <html>");
<PRINTLN>(" <form>");
<PRINTLN>(" <Table  bgcolor=gold>");
  <PRINTLN> ("ATM customers...");
<PRINTLN> ("<BR>**********");
string[] add = new string[lengthA];
string  d11=armg.get(0).ToString();
string  d1=armg.get(1).ToString();
for  (int  i=0;i<lengthA;i+=1)
{
 <PRINTLN>(" <td>");
```

```
add[i]=ename[i].Replace("%2F","/").Replace("%40","@").Replace("+"," ").Replace(","," ").ToString();
tos.Add(add[i]);
            <PRINTLN>(" </td>");
<PRINTLN>("<BR>");
 }
<PRINTLN>(" </table>");
<PRINTLN>(" </form>");
<PRINTLN>(" </html>");
string amtn="";
 foreach (var arr in tos)
          {
              amtn = string.Format("{0}", arr);
          }
          int num1=Int32.Parse(d1)+Int32.Parse(amtn);
<PRINTLN>("amount1="+amtn);
<PRINTLN>("amount2="+d1);
//updte  the  atmtranscation
          String s171 = "UPDATE from atmtranscation 5 to 18 , 1 to 5 ?= C By 1 1 : {"+amtn+"}
:{"+""+num1+"}:{0}";
            wdbaconn.WDBAQUERY(s171);
<PRINTLN>(" <BR><a href=/index.BS class=active >Click here ATM MENU</a>");
    Environment.Exit(-1);
}
<CATCH>(<EXE> e)
{
}
}
}
}
```

JDollar Program to Withdraw the AMount

```
<JDollar>
<USE> <WEB>.util;
<USE> Security;
```

```
<PACK> Program5
{
     <CLASS> Prog
  {
public static ArrayList tos= new ArrayList();
       public void Main()
     {
ArrayList arm1= new ArrayList();
arm1.add("idno2");
arm1.add("amt2");
arm1.add("NOT");
<PRINTLN>("<HTML>");
<PRINTLN>("<BODY bgcolor=pink>");
<PRINTLN>("<form>");
ArrayList  armg= Request.Query(arm1,"withdraw.BS.dsn",2,1);
wdbaconn.JSTARWDBAQUERY("datastores", "USEDATABASE", "dbpwds",
"C:\\Programs\\WNOSQL\\WNOSQLProgramfiles\\WNOSQL-cod");
 wdbaconn.JSTARWDBAUSERQUERY("dbuser", "dbpwds", "wilmix78", "wilmix78");
String  qh2="SELECTRVAL from atmtranscation 5 to 10 , 1 to 5 ?= 1 By 1 1 : {0} : {0} :{0}";
  wdbaconn.WDBAQUERY(qh2);
 int  n=0;int  xx=0;int  amts=0;int totalamt=0;
<TRY>
{
String s=Secure.RetreiveSecure("output.wdba" ,0); //retrieve  the  query  output   from   wdba file
s=s.Replace("[","").Replace("]","");
string []ename = s.Split(' ');
    int lengthA = ename.Length;
<PRINTLN>(" <html>");
<PRINTLN>(" <form>");
<PRINTLN>(" <Table  bgcolor=gold>");
  <PRINTLN> ("ATM customers...");

<PRINTLN> ("<BR>**********");

string[] add = new string[lengthA];

string  d11=armg.get(0).ToString();
```

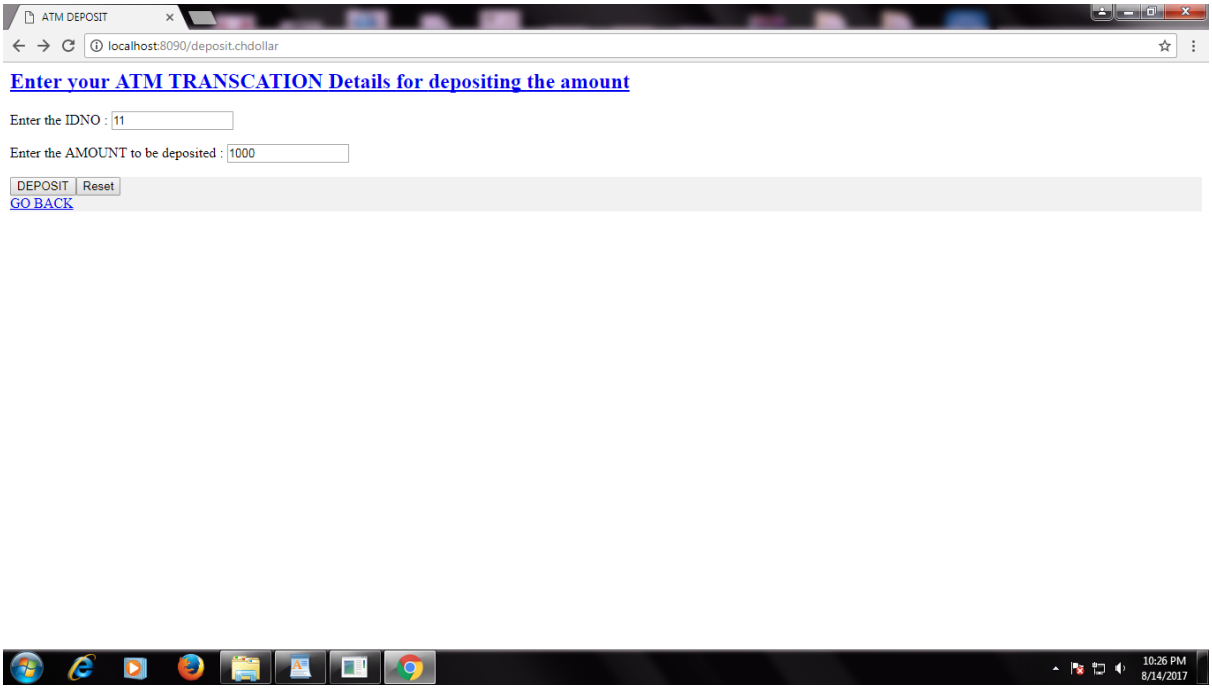
```
string  d1=armg.get(1).ToString();

for  (int  i=0;i<lengthA;i+=1)
{
 <PRINTLN>(" <td>");
add[i]=ename[i].Replace("%2F","/").Replace("%40","@").Replace("+"," ").Replace(","," ").ToString();
tos.Add(add[i]);

                    <PRINTLN>(" </td>");
<PRINTLN>("<BR>");

 }
<PRINTLN>(" </table>");

<PRINTLN>(" </form>");
<PRINTLN>(" </html>");
string amtn="";
 foreach (var arr in tos)
          {
              amtn = string.Format("{0}", arr);
          }
          int num1=Int32.Parse(amtn)-Int32.Parse(d1);
<PRINTLN>("amount1="+amtn);
<PRINTLN>("amount2="+d1);
          String s171 = "UPDATE from atmtranscation 5 to 18 , 1 to 5 ?= C By 1 1 : {"+amtn+"}
:{"+""+num1+"}:{0}";
            wdbaconn.WDBAQUERY(s171);
<PRINTLN>(" <BR><a href=/index.BS class=active >Click here ATM MENU</a>");
   Environment.Exit(-1);

}

<CATCH>(<EXE> e)

{
}
```

```
}

}

}
```

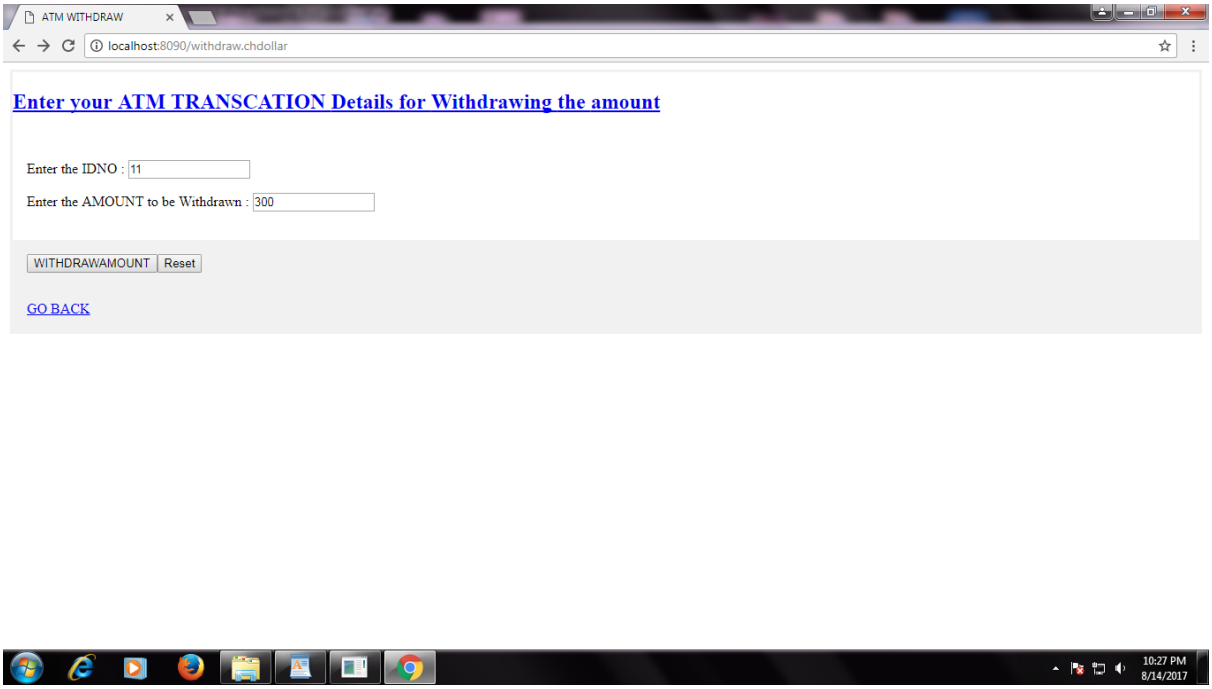
OUTPUTS

Note: Outputs are given in next page for user convenience

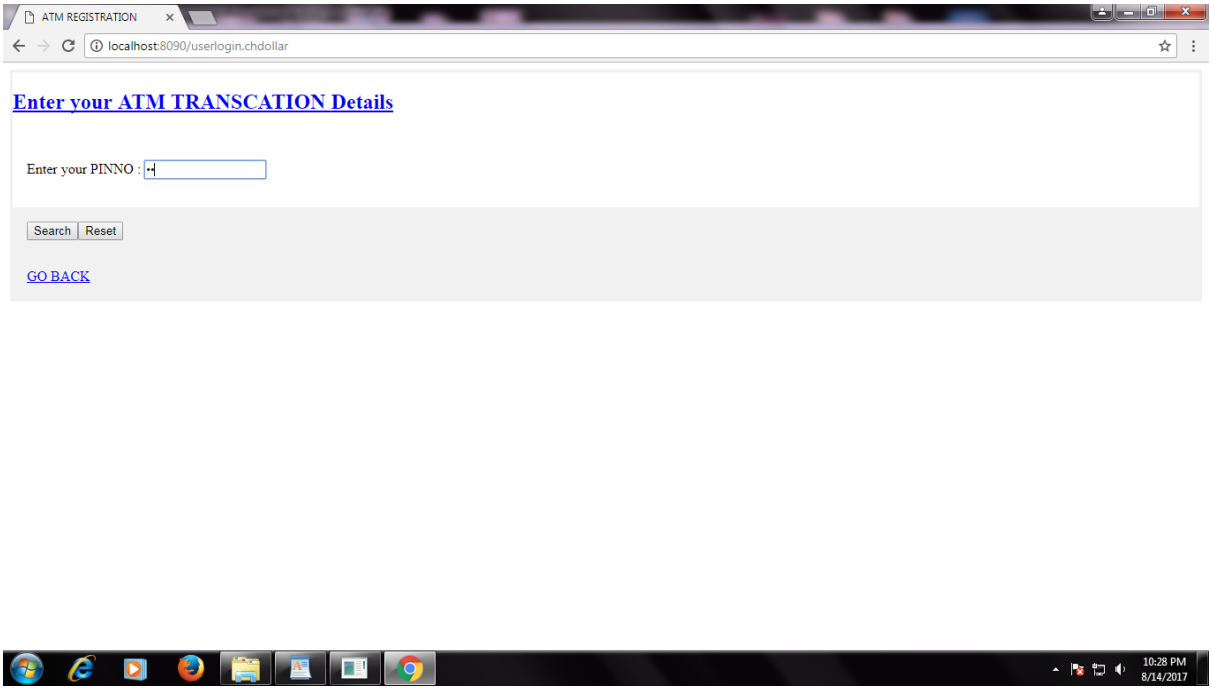
A) Depositing the Amount



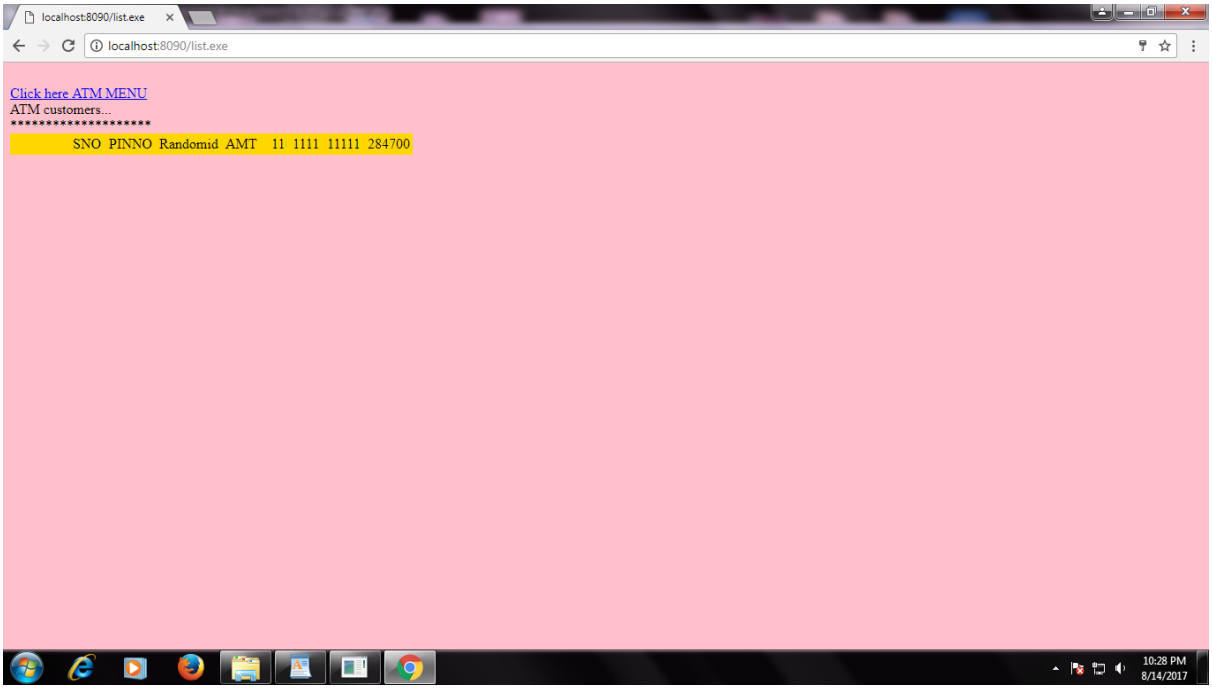
B) WithDrawing the Amount



C) Check the Atm amt



d) list the balance details for the customer



==========

Unit -9: CHDollar with JWP

==========

CHdollar with JWP is focused on Chdollar with Jdollar interaction.

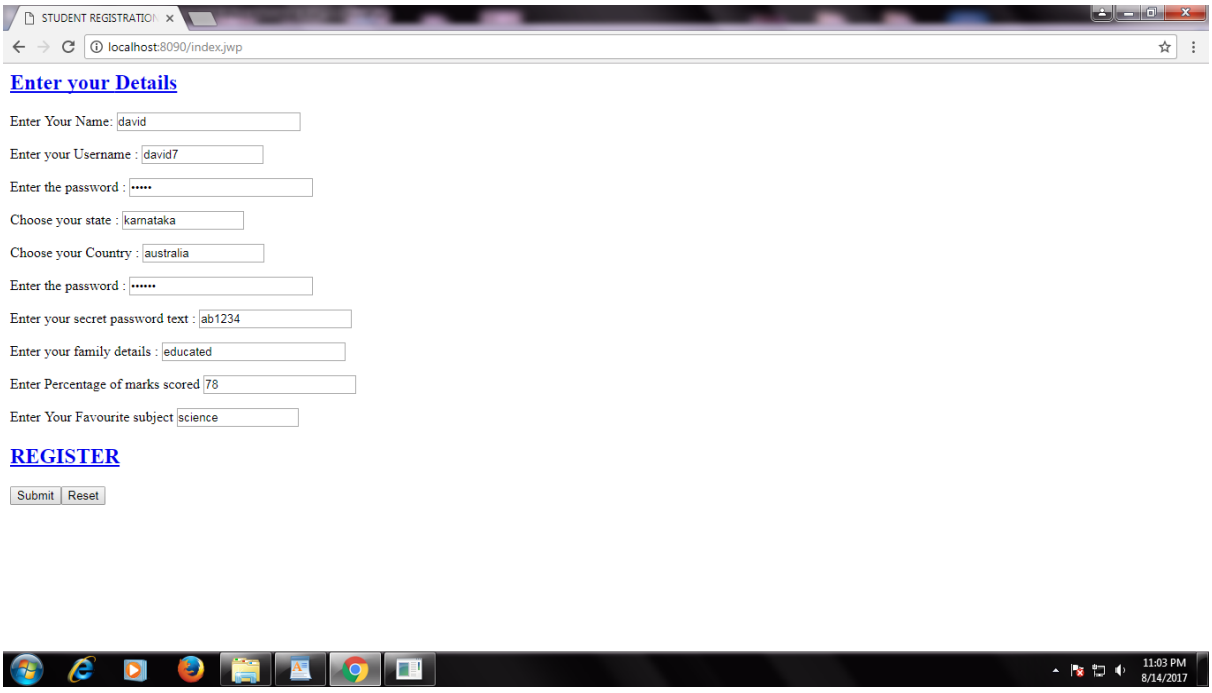
SYNTAX:

```
<JDWEB>
<PACK>
<%
public class index {
<J$MISC>
<! logic !>
</J$MISC>

 public void  JWP-Main( )
{
<! JWP  Logic  !>
}
}
%>
</JDWEB>
```

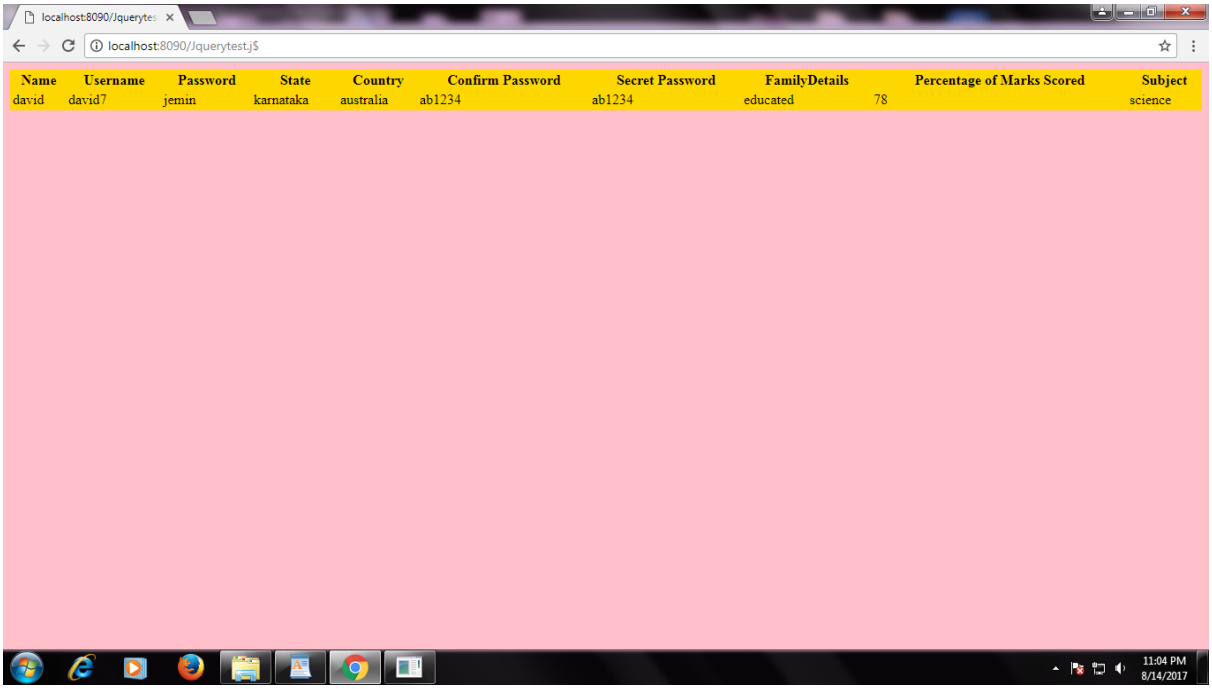
Example-1

```
 <JDWEB>

<PACK>
<%

public class index {

<J$MISC>
form.put("wil","11");
</J$MISC>

 public void  JWP-Main( )
{
HTML.displayhtml("Register.html");
<HMAP> mp = <NEW> <HMAP>();
mp.PUT("111",776);
JDollar.Writeln(""+MISCBean(mp));
}
}
%>
</JDWEB>
```

Input


STUDENT REGISTRATION
localhost:8090/index.jwp
Enter your Details
Enter Your Name: david
Enter your Username : david7
Enter the password : •••••
Choose your state : karnataka
Choose your Country : australia
Enter the password : ••••••
Enter your secret password text : ab1234
Enter your family details : educated
Enter Percentage of marks scored 78
Enter Your Favourite subject science
REGISTER
Submit
Reset
11:03 PM
8/14/2017

Output:



FAQS?

Why we use the following code in .j$ which is given below?

.....

.....

.....

<J$MISC>

<! logic !>

</J$MISC>

.....

.....

.....

Since <J$MISC> indicates other domains like Reservation,etc.

==========

UNIT-10: CH$ with CDollar ,JDollar oops and GDollar oops concepts

==========

JDollar OOPS like Linear List ,ArrayLinearList,etc.
And CDollar oops like Bag dts is followed for tree datastrucures.

And GDollar oops like BOOKS,LOOPS,UNION,and ,USERCASE is followed for CHDollar.

Shopping cart using CDollar dts

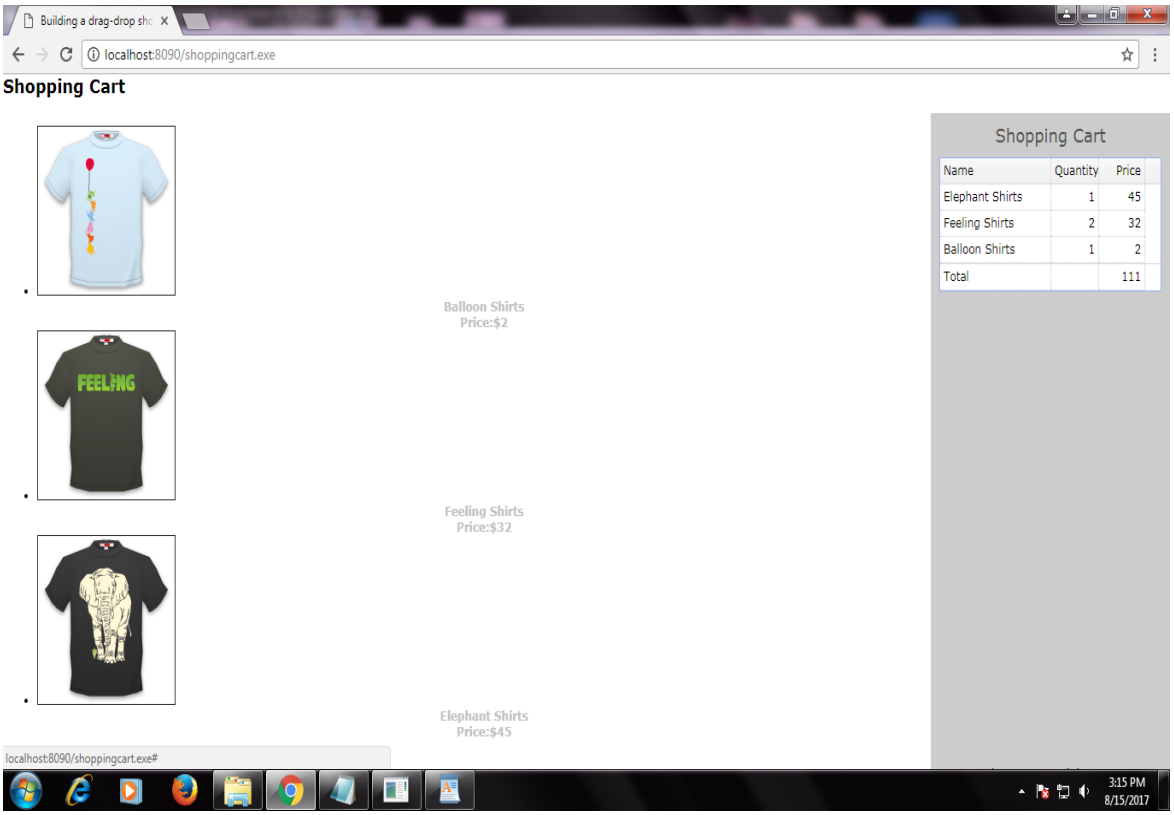
Example for Shopping cart

```
<JDollar>
<PACK> Program5
{
   <CLASS> Prog
  {
     public void Main()
     {

 Bag  Member <NEW> Bag();

Member.PUT("A","Price:$2");
Member.PUT("B","Price:$32");
Member.PUT("C","Price:$45");
Member.PUT("D","Price:$12");
Member.PUT("E","Price:$15");
Member.PUT("F","Price:$15");
//put price values in A,B,C,D,E,F
// and get the value using Member.get("A") and so-on
```

```
HTML.displayhtml("shoppingcart.html");
<PRINTLN>("<div class=products>");
                <PRINTLN>("<ul>");
                        <PRINTLN>("<li>");
                                <PRINTLN>("<a href=# <CLASS>=item>");
                                        <PRINTLN>("<img
src=http://www.jeasyui.com/tutorial/dd/images/shirt1.gif>");
                                        <PRINTLN>("<div>");
                                        <PRINTLN>("<p>Balloon Shirts</p>");

        <PRINTLN>("<p>"+Member.get("A").ToString().Replace("[","").Replace("]","")+"</p>");
                                        <PRINTLN>("</div>");
                                <PRINTLN>("</a>");
                        <PRINTLN>("</li>");
                        <PRINTLN>("<li>");
                                <PRINTLN>("<a href=# <CLASS>=item>");
                                        <PRINTLN>("<img
src=http://www.jeasyui.com/tutorial/dd/images/shirt2.gif>");
                                        <PRINTLN>("<div>");
                                                <PRINTLN>("<p>Feeling Shirts</p>");

        <PRINTLN>("<p>"+Member.get("B").ToString().Replace("[","").Replace("]","")+"</p>");
                                        <PRINTLN>("</div>");
                                <PRINTLN>("</a>");
                        <PRINTLN>("</li>");
                        <PRINTLN>("<li>");
                                <PRINTLN>("<a href=# <CLASS>=item>");
                                        <PRINTLN>("<img
src=http://www.jeasyui.com/tutorial/dd/images/shirt3.gif>");
                                        <PRINTLN>("<div>");
                                                <PRINTLN>("<p>Elephant Shirts</p>");

        <PRINTLN>("<p>"+Member.get("C").ToString().Replace("[","").Replace("]","")+"</p>");
                                        <PRINTLN>("</div>");
                                <PRINTLN>("</a>");
                        <PRINTLN>("</li>");
```

```
                <PRINTLN>("</ul>");
            <PRINTLN>("</div>");

HTML.displayhtml("shoppingcart1.html");
}
}

}
```

Output:



==========

UNIT-11: CH$ MOCK EXERCISES

==========

CH$ Mock Exercises:

Section-A

(1 *100 =100 marks)

A) Write a CH$ Program to design a BANK Remote Web Application -TYPE-2 (1* 10 = 25 marks)

with transcations credit,debit,withdraw cash, discount ,etc.

B) Write a CH$ Program for Electricity online Bill? (1*5 = 5 marks)

c) Write a CH$ program with JQUERY to build a tree structure in CH$ webpage

(1*10 = 10 marks)

d) Write a CH$ Program to build a webapplication to enter all

student details in a form and store it using wnosql database.

after that update it ,retrieve it and print the webpage.

(1*10=10 marks)

e) Write a CH$ program with JQUERY or bootstrap to list the contents

from wnosql database and print it in table format. (1*5= 5 marks)

f) Write a CH$ MVC program (1*20 =20 marks)

to check whether the student name present or not from student form with wnosql

and create fields using wnosql db with name,course,dateofjoin,dateoffinish, and status.

and perform logic in model class

like

```
if (course=="java")
amt="2000"
else
if (course=="c/c++")
amt="10000"
else
if (course=="dotnet")
amt="15000"
else
if (course=="php")
amt="5000"
else
if (course=="mgt")
amt="25000"
```

if the student did not payed the fees mark the status as

"unpaid" otherwise mark the status "paid".

after that list all paid and unpaid people in

a seperate webpage.

g) Write briefly about the workflow of CH$ Program.(1 *5=5 marks)

H) Develop a ATM Transcation Project using

CH$ Remote Webapplication -2. (1 * 20 = 20 marks)

SECTION -B

(4 *50 =200 marks)

A) Write a CH$ Program for Biometric web application to maintain attendance system using CH$-JSTAR

(1 *50=50 marks).

b) Write A CH$ -J$ Program for creating library management system.(1 *50 =50 marks)

c) How Linear datastructure for Jdollar(JWEB) works? What is the use of using Linear Datastructure ? Write a CH$ program to Display student details with fields (CLASS NO , STUDENTNAMES) (1 * 50 = 50 marks)

eg) if you provide a data in a non-sequence order.

1, jackie

0,dion

2,suresh

9, jackie

3,suresh1

4,suresh2

5,suresh3

6,suresh4

7,suresh5

8,suresh6

Expected output will be:

0,dion

1, jackie

2,suresh

3,suresh1

4,suresh2

5,suresh3

6,suresh4

7,suresh5

8,suresh6

9, jackie

d) Write a CHDollar with JWP to develop a Bus reservation system... (1 * 50=50 marks)

UNIT 12: Brief Fundamentals of CHDollar

==========

FUNDAMENTALS of CHDOLLAR

==========

DATASTRUCTURES for CHDollar

==========

A) CHDollar APIS

==========

Desktop.Screen(<string>); => Used to create a RemoteDesktop

TEMPLATE.style(String sop) => used to print any template of html , style sheet ,etc

FingerPrint.call(<path>); => it is used to generate a fingerprint no

B) ATM APS

=========

ATM APIS => it is used for credit , debit, fraction (discount,interest,etc) purpose with Arraylist as Build-in functions

ATM.CREDIT(int amt,int cr)

ATM.DEBIT(int amt,int cr)

ATM.FRACTION(int amt,int cr,int fact)

ATM.ArrayList

C)WITHRECORD

==========

WithRecord also act as a Book dts but

it is used to handle large no of records with Withrecord-arraylist

eg)

```
WithRecord u = new WithRecord("Wilmix");

for (int i=0;i<=100;i++)
u.add("col"+i);

ArrayList ar = new ArrayList();
for (int j=0;j<=10000;j++)
ar.add(j,j);

u.RECORDA("0001" , ar.ToString());

<PRINTLN>(""+u.rett1(1));
```

<PRINTLN>(""+u.rett2(1));

u.RECORDB("Physics" , ar.ToString());

<PRINTLN>(""+u.rett11(1));

<PRINTLN>(""+u.rett12(1));

D)Dictionary with ArrayList

==========

It is used to make search and insertion as fast as possible..

E)GRAPH DTS

==========

a) DirectedGraph

==========

This is used to test whether the Graph contains cycle or not.....

and it is used to perform insertion and search mechanism.

```
DirectedGraph a = new DirectedGraph(5);
        a.addEdge(1, 0);
        a.addEdge(0, 21);
        a.addEdge(21, 0);
        a.addEdge(0, 31);
        a.addEdge(31, 4);

a.add(0,1);
a.add(0,100);

        if (a.isCyclic())
            <PRINTLN>("contains cycle");
        else
            <PRINTLN>("No cycle");
```

b) DepthFirstsearchTraversal

==========

This will perform depthfirstsearch starting from vertex a and it is used to perform insertion and search mechanism.

```
GraphDFS a1 = new GraphDFS(4);

    a1.addEdge(0, 11);
    a1.addEdge(0, 21);
    a1.addEdge(11, 21);
    a1.addEdge(21, 0);
    a1.addEdge(21, 31);
    a1.addEdge(31, 31);

    a1.DFS(21);
```

Note: This will perform depthfirstsearch starting from vertex 21

==========

F) CHDollar Functions

==========

To execute only .exe files use RunnabletimeExec.call(exefilename);

APP.display() => This is used to invoke EMAIL.exe

APP.insert(Object obj,String path) => you can insert any arraylist object to a filewith pathname

==========

G) WXML f(x)

==========

This is used to build Table , report, form ,bill...

h) CHDollar Optimized DTS

==========

BOOK ,USECASE,UNION,LOOP are the Optimized DTS

BOOK

=====

It is used to store and retrieve large amount of records

with powerful insertion and search mechanism

LOOP

====

Consider a scenario you are given a 1000000000 (1 million) records

which is retrieved form sql /oracle/wnosql database..

You want to write a code in java or C# ...

This will definetly make your memory trouble shoot and very slow...

Finally it is impossible ....

This Loop dts divide the data into batches and

store it in different objects ....

and this prevents memory slow..

Here you can easily search the data from LOOP

with powerful insertion and search mechanism like ArrayList.

eg)

```
LOOP u = new LOOP("Wilmix");

for (int i=0;i<=100;i++)
u.add("col"+i);
LOOP u1 = new LOOP("Wilmix");

for (int i=7;i<=107;i++)
u.LOC1("col"+i);
```

```
LOOP u2 = new LOOP("Wilmix");

for (int i=5;i<=106;i++)
u.LOC2("col"+i);

<PRINTLN>(""+u.match("col5",5));

<PRINTLN>(""+u.matchu1("col5",5));

<PRINTLN>(""+u.matchu2("col5",5));
```

UNION in CHDollar

==========

Like C datastructure UNION, CHDollar structures UNION

will saves memory space , time and cost...

UNION in CHDollar has in build arraylist

for powerful insertion and search mechanism.

eg)

```
UNION u = new UNION("Wilmix");

for (int i=0;i<=100;i++)
u.add("col"+i);

<PRINTLN>(""+u.match("col5",5)); // will test whether col5 is at location 5 and it returns a boolean value....
```

USECASE in CHDollar

==========

USECASE dts is used to store values , used to store values at LOC1

or LOC2....

and it is used to search a 5 value at 5th location at LOC1 and LOC2.

```
USECASE <Integer> u = new USECASE<Integer>(0);
int i=0;
```

```
for ( i=0;i<=100;i++)
{
u.add(i);
u.LOC1(i+100);
u.LOC2(i+200);
}
<PRINTLN>(""+u.match(5,5));

<PRINTLN>(""+u.matchu1(5,5));


<PRINTLN>(""+u.matchu2(5,5));


<PRINTLN>(""+u.switchcase('+', 10,1,20,20,40));
```

Output

======

```
True
False
False
40
```

<PRINTLN>(""+u.switchcase('+', a,a1,a2,a3,a4));

It is a shortcut statement for switch case statement

so it optimize your source code...

> => a

+ => a4

= => a3

- => a2

< => a1

that means if you assign > symbol it will

assign a value say a in this eg) 10

CHDollar with C#

==========

CHDollar is also an interative technology and it is also used with

C#.

Sample-1

=========

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using chdollarlib;
namespace ConsoleApplication1
{
   class Program
   {
      static void Main(string[] args)
      {

         BOOK b = new BOOK();
```

```
            b.add(0, 11);
            b.add(0, -1);

            Console.WriteLine("" + b.get(0));
 Console.WriteLine("" + b.get(1));



        }
    }
}
```

Output

======

F:\ConsoleApplication1>consoleapplication1

-1

11

Note: Book datastructure in chdollar is also act as a arraylist.

=====

CHDollar Webpage

==========

example-1

=========

```
<CHSAUCER>

<USE> jstarlib;//load  jstarlib  packages
<PACK>  p
{
  <CLASS> index
  {
    public void main()
    {



HTML.displayhtml("Register.html"); //  it  will  display  the  contents  to  webpage
```

```
}

}
}
```

```
</CHSAUCER>
```

Output:

========

It will display Registerationwebpage

example-2

=========

```
<CHSAUCER>
<USE> jstarlib;
```

```
<PACK>  p
{
  <CLASS> MyArray
  {
    public void main()
    {



HTML.displayhtml("ang.html");// it will  display  the contents to webpage
}


}


}


</CHSAUCER>
```

example-3

==========

```
<CHSAUCER>
```

```
<USE> jstarlib;
<PACK>  p
{
  <CLASS> MyArray
  {
    public void main()
    {




HTML.displayhtml("animate.html");
}


}


}


</CHSAUCER>
```

example-4

=========

CHDollar Book datastructures

======================

Difference between CHDollar book dts and Gdollar book dts

GDollar book dts did not have in build arraylist

but CHDollar have...

<CHDollar>

<USE> chdollarlib;

<PACK> MyP

<%

<CLASS> Programs

<%

public FLOAT CHDollar-MAIN()

```
<%


BOOK u <NEW> BOOK("Wilmix");

for (int i=0;i<=100;i++)
{
u.add("col"+i);
u.add(0,"col"+i);

}

<PRINTLN>(""+u.match("col5",5));
u.STOREA("0001" , u.ToString());

<PRINTLN>(""+u.rett1(1));
<PRINTLN>(""+u.rett2(1));

u.STOREB("Physics" , u.ToString());

<PRINTLN>(""+u.rett11(1));
<PRINTLN>(""+u.rett12(1));
```

```
%>


?>
```

Output:

```
=======

True

0001

[col100, col99, col98, col97

, col96, col95, col94, col93

, col92, col91, col90, col89

, col88, col87, col86, col85

, col84, col83, col82, col81

, col80, col79, col78, col77

, col76, col75, col74, col73

, col72, col71, col70, col69

, col68, col67, col66, col65

, col64, col63, col62, col61

, col60, col59, col58, col57

, col56, col55, col54, col53

, col52, col51, col50, col49

, col48, col47, col46, col45
```

, col44, col43, col42, col41

, col40, col39, col38, col37

, col36, col35, col34, col33

, col32, col31, col30, col29

, col28, col27, col26, col25

, col24, col23, col22, col21

, col20, col19, col18, col17

, col16, col15, col14, col13

, col12, col11, col10, col9,

col8, col7, col6, col5, col

4, col3, col2, col1, col0]

[col100, col99, col98, col97

, col96, col95, col94, col93

, col92, col91, col90, col89

, col88, col87, col86, col85

, col84, col83, col82, col81

, col80, col79, col78, col77

, col76, col75, col74, col73

, col72, col71, col70, col69

, col68, col67, col66, col65

, col64, col63, col62, col61

, col60, col59, col58, col57

, col56, col55, col54, col53

, col52, col51, col50, col49

, col48, col47, col46, col45

, col44, col43, col42, col41

, col40, col39, col38, col37

, col36, col35, col34, col33

, col32, col31, col30, col29

, col28, col27, col26, col25

, col24, col23, col22, col21

, col20, col19, col18, col17

, col16, col15, col14, col13

, col12, col11, col10, col9,

col8, col7, col6, col5, col

4, col3, col2, col1, col0]

0001

CHDollar AJAX

==========

Program-1

==========

```
<CHSAUCER>


<PACK>  p
{
  <CLASS> MyArray
  {
    public void main()
    {









<PRINTLN>("<html>");
<PRINTLN>("<head><title>Welcome   to  Online BILLING  </title></head>");
<PRINTLN>("<center> <U> Welcome   to  Online BILLING  </u> </center>");
<PRINTLN>("<body bgcolor=blue> ");
<PRINTLN>("<TABLE  bgcolor=pink> ");
<PRINTLN>("<TH>Welcome   to  ONLINE BILLING  WEBPAGE <TH>");
<PRINTLN>("<TD> ");


<PRINTLN>( "</td>");
<PRINTLN>("</TABLE>");



<PRINTLN>("<form action=BILL.CH$ method=post>");
```

```
 <PRINTLN>(" <label for=id>ID</label>");
  <PRINTLN>("<input name=id />");
  <PRINTLN>("<br/>");
  <PRINTLN>("<label for=summary>BILLDETAILS</label>");
 <PRINTLN>(" <input name=summary />");
  <PRINTLN>("<br/>");
  <PRINTLN>("Description:");
 <PRINTLN>(" <TEXTAREA NAME=description COLS=40 ROWS=6></TEXTAREA>");
  <PRINTLN>("<br/>");
 <PRINTLN>("AMT:");
 <PRINTLN>(" <input type=text name=t1></TEXTAREA>");

  <PRINTLN>("<input type=submit value=Submit />");
 <PRINTLN>(" </form>");
<PRINTLN>("</html>");



}


}


}


</CHSAUCER>
```

Note: This program redirects to the same webpage

=====

Program-1

=========

```
<CHSAUCER>
<USE> jstarlib;
<PACK>  p
{
  <CLASS> MyArray
  {
    public void main()
    {



RuntimeExec.call("animate");
}

}

}
```

</CHSAUCER>

note: This will display the animate.exe contents to the browser...

CHDollar with CDollar

==========

// this is my CHSAUCER first program

<CHSAUCER>

<USE> CUTIL;

<PACK> p

{

```
<CLASS> MyArray
{
   public void main()
   {
      int []  n = new int[10]; /* n is an array of 10 integers */
      int i,j;

ArrayList  ar  =  new  ArrayList();

      /* initialize elements of array n */
      for ( i = 0; i < 10; i++ )
      {
        n[ i ] = i + 100;

ar.add(i,n[i]);

      }

      /* output each array element's value */
      for (j = 0; j < 10; j++ )
      {
        <PRINTLN>(""+j,""+ar.get(j).ToString());
```

```
        }


      }
    }
}

</CHSAUCER>
```

Output:

=======

0123456789

CHDollar with terminal Desktop

==========

Desktop.chdollar

==========

```
<CHDollar>

<USE>  chdollarlib;



<PACK> MyP
<%
   <CLASS> Programs
   <%
        public FLOAT CHDollar-MAIN()



      <%


Desktop.Screen("GalaxycloudDesktop");



%>



?>
```

Note: This will create Desktop.exe using CWE-Editor

==========

<CHSAUCER>

<USE> chdollarlib;

<PACK> p

{

<CLASS> MyArray

{

public void main()

{

RuntimeExec.call("Desktop");

}

}

}

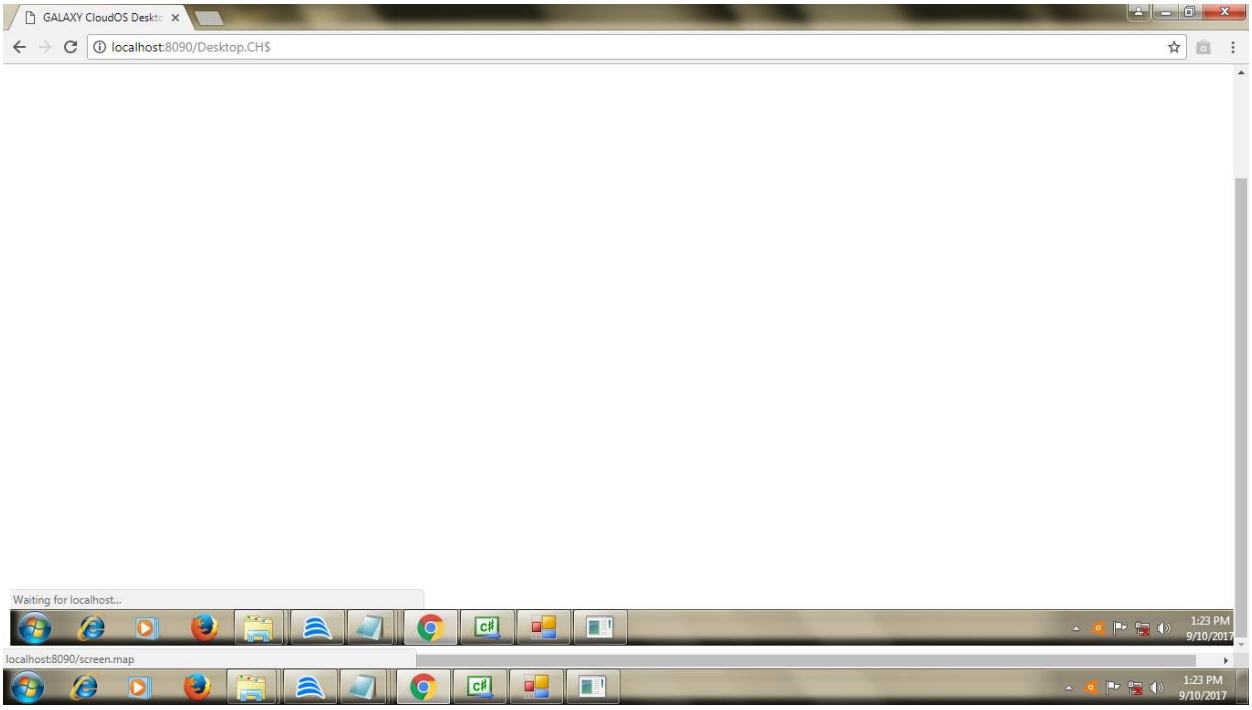
</CHSAUCER>

Output

======

Displays a Terminal desktop to be viewed in mobiles... and

Display which icons are open in ur desktop.



==========

CHDollar Reference Documentation

==========

CHDOLLAR Latest Tutorial will be updated in the given below Websites.....

kindly go thru given tutorial url for more details....

==========

==========