BridgeLabz

Employability Delivered

Programming Constructs - Arrays

5. Arrays

BridgeLabz
Employability Delivered

An array is a systematic arrangement of the same type of data.

But in Shell script Array is a variable which contains multiple values may be of same type or different type since by default in shell script everything is treated as a string.

An array is zero-based i.e. indexing start with 0.

Array Example

```
#!/bin/bash -x

couter=0
Fruits[((counter++))]="Apple"
Fruits[((counter++))]="Banana"
Fruits[((counter++))]="Orange"

echo ${Fruits[@]}
arrayTest.sh (END)
```

```
+ couter=0
+ Fruits[((counter++))]=Apple
+ Fruits[((counter++))]=Banana
+ Fruits[((counter++))]=Orange
+ echo Apple Banana Orange
Apple Banana Orange
```

UC 8

Store the Daily Wage along with the Total Wage

Storing Daily Wage in Array

```
#!/bin/bash -x

# CONSTANTS FOR THE PROGRAM
IS_PART_TIME=1;
IS_FULL_TIME=2;
MAX_HRS_IN_MONTH=10;
EMP_RATE_PER_HR=20;
NUM_WORKING_DAYS=20;

# VARIABLES
totalWorkHours=0;
totalWorkingDays=0;

function getWorkingHours() {
   case $1 in
      $IS_FULL_TIME)
         workHours=8
         ;;
      $IS_PART_TIME)
         workHours=4
         ;;
      *)
         workHours=0
         ;;
   esac
   echo $workHours
}

function calcDailyWage() {
   local workHrs=$1
   wage=$(($workHrs*$EMP_RATE_PER_HR))
   echo $wage
}

while [[ $totalWorkHours -lt $MAX_HRS_IN_MONTH &&
         $totalWorkingDays  -lt  $NUM_WORKING_DAYS ]]
do
   ((totalWorkingDays++))
   workHours="$( getWorkingHours $((RANDOM%3)) )"
   totalWorkHours=$(($totalWorkHours+$workHours))
   empDailyWage[$totalWorkingDays]="$( calcDailyWage $workHours )"
done

totalSalary="$( calcDailyWage $totalWorkHours )"
echo "Daily Wage " ${empDailyWage[@]}
```

```
+ IS_PART_TIME=1
+ IS_FULL_TIME=2
+ MAX_HRS_IN_MONTH=4
+ EMP_RATE_PER_HR=20
+ NUM_WORKING_DAYS=20
+ totalWorkHours=0
+ totalWorkingDays=0
+ (( 0 -lt 4 ))
+ (( 0 -lt 20 ))
+ (( totalWorkingDays++ ))
++ getWorkingHours 0
++ case $1 in
++ workHours=0
++ echo 0
+ workHours=0
+ totalWorkHours=0
++ calcDailyWage 0
++ local workHrs=0
++ wage=0
++ echo 0
+ empDailyWage[$totalWorkingDays]=0
+ (( 0 -lt 4 ))
+ (( 1 -lt 20 ))
+ (( totalWorkingDays++ ))
++ getWorkingHours 0
++ case $1 in
++ workHours=0
++ echo 0
+ workHours=0
+ totalWorkHours=0
++ calcDailyWage 0
++ local workHrs=0
++ wage=0
++ echo 0
+ empDailyWage[$totalWorkingDays]=0
+ (( 0 -lt 4 ))
+ (( 2 -lt 20 ))
+ (( totalWorkingDays++ ))
++ getWorkingHours 2
++ case $1 in
++ workHours=8
++ echo 8
+ workHours=8
+ totalWorkHours=8
++ calcDailyWage 8
++ local workHrs=8
++ wage=160
++ echo 160
+ empDailyWage[$totalWorkingDays]=160
+ (( 8 -lt 4 ))
++ calcDailyWage 8
++ local workHrs=8
++ wage=160
++ echo 160
+ totalSalary=160
+ echo 'Daily Wage ' 0 0 160
```

BridgeLabz
Employability Delivered

Arrays Practice Problems

BridgeLabz
Employability Delivered

1. Write a program that does the following
   a. Generates 10 Random 3 Digit number.
   b. Store this random numbers into a array.
   c. Then find the 2nd largest and the 2nd smallest element without sorting the array.
2. Extend the above program to sort the array and then find the 2^{nd} largest and the 2^{nd} smallest element.
3. Extend the Prime Factorization Program to store all the Prime Factors of a number n into an array and finally display the output.
4. Write a Program to show Sum of three Integer adds to ZERO
5. Take a range from 0 – 100, find the digits that are repeated twice like 33, 77, etc and store them in an array

BridgeLabz

Employability Delivered

Thank You