Pseudo code

```
$daysbeforeexpirytonotify = 14
$now = (get-date).ToUniversalTime().ToFileTime()
$threshold = (get-date).ToUniversalTime().adddays($daysbeforeexpirytonotify).ToFileTime()
$MaxPwdAge = (Get-ADDefaultDomainPasswordPolicy).MaxPasswordAge.Days
$expiredDate = (Get-Date).addDays(-$MaxPwdAge)
$maxdays=90
$FileLogdate = (Get-Date -f dd_MM_yyyy_HH_mm_ss)
$File_email = "C:\dir1\EmailSent_$FileLogdate.txt"
$File_error = "C:\dir2\ErrPswdExpNew_$FileLogdate.txt"

write output "MaxpwdAge = $MaxPwdAge"
write output "ExpiredDate = $expiredDate"
write-output $FileLogdate
write-output "Writtten to Email File  $File_email"
write-output "Writtten to Email File  $File_error"

"User ID | User Name   | User EmailAddress | Expiry Date | Date Email Sent | Comments" |
Out-File -FilePath $File_email
"User ID | User Name   | User EmailAddress | Date Email Sent | Expirydate  | Daystoexpiry |
Comments " | Out-File -FilePath $File_error

$users = Get-ADUser -filter {  (Enabled -eq $True) -and (PasswordNeverExpires -eq $False)} -
     properties Name, PasswordNeverExpires, PasswordExpired, PasswordLastSet, EmailAddress
     ,msDS-UserPasswordExpiryTimeComputed -SearchBase "DC=<DCName>,DC=com" |# where {
     $_."msDS-UserPasswordExpiryTimeComputed" -lt $threshold -and $_."msDS-
     UserPasswordExpiryTimeComputed" -gt $now } |
   Select-Object -Property  "Name",
            "EmailAddress",
            "SamAccountName",
            "PasswordLastSet",
            @{Name="ExpiryDate";Expression={
                [datetime]::FromFileTime($_."msDS-UserPasswordExpiryTimeComputed")
                }
            },
            @{Name="DaysToExpiry";Expression={
                 [int](($_."msDS-UserPasswordExpiryTimeComputed" - $now) /
                  864000000000)
                 }
            }
```

```
$usersCount = ($users | Measure-Object).Count
Write-Output "Found $usersCount Users"
Write-Output "Found $usersCount User Objects"
$users

foreach ($user in $users) {
  if ( $user.DaysToExpiry -lt 1) {
      if (($user.daystoexpiry -ge -$maxdays) -and ($user.daystoexpiry -le -1)){
          "$($user.SamAccountName) | $($User.Name)   | $($User.EmailAddress) |
           $FileLogdate
           |$($user.expirydate)  | $($user.daystoexpiry) | The Password for this user has expired..
           Sending an email to the user" | Out-File -FilePath $File_email -Append
          send-mailmessage -from "sample@abc.com" -To $user.EmailAddress -smtpserver
          < SMTPIP >-subject "Your  password has expired $($user.daystoexpiry) days Ago" –
          body "Hello $($user.name), `n Your  password Has Expired on $($user.expirydate)
          (UTC). Please Create a  Incident ticket and assign it to Support group! `n `n Thanks, `n
          Support Team `n support@abc.com" -Attachments "C:\dir1\Instructions for Updating
          Passwords.pdf"
                }
      else {
         "$($user.SamAccountName) | $($User.Name)   | $($User.EmailAddress) | $FileLogdate
          |$($user.expirydate)  | $($user.daystoexpiry) | There is something wrong with this
          record..   Please Validate" | Out-File -FilePath $File_error -Append
          write-output "Bad Date .. Please check.. $user.name"
    }
    }
   else {
      if ($user.DaysToExpiry -le 14) {
          send-mailmessage -from "sample@abc.com" -To $user.EmailAddress -smtpserver
          < SMTPIP >-subject "Your  password will expire in $($user.daystoexpiry) days" -body
          "Hello $($user.name),`n `n Your  password will expire at $($user.expirydate) (UTC).
          Please see the attached Instructions.pdf document on how to reset your password! `n `n
          Thanks, `n Support Team `n support@abc.com" -Attachments "C:\dir1\Instructions for
          Updating Passwords.pdf"

    }
    }
  }

#------------------ END SCRIPT ---------------------------------------------
```