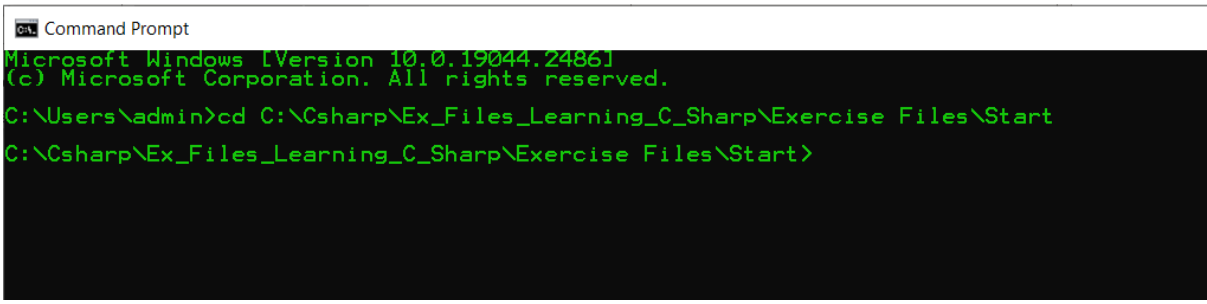
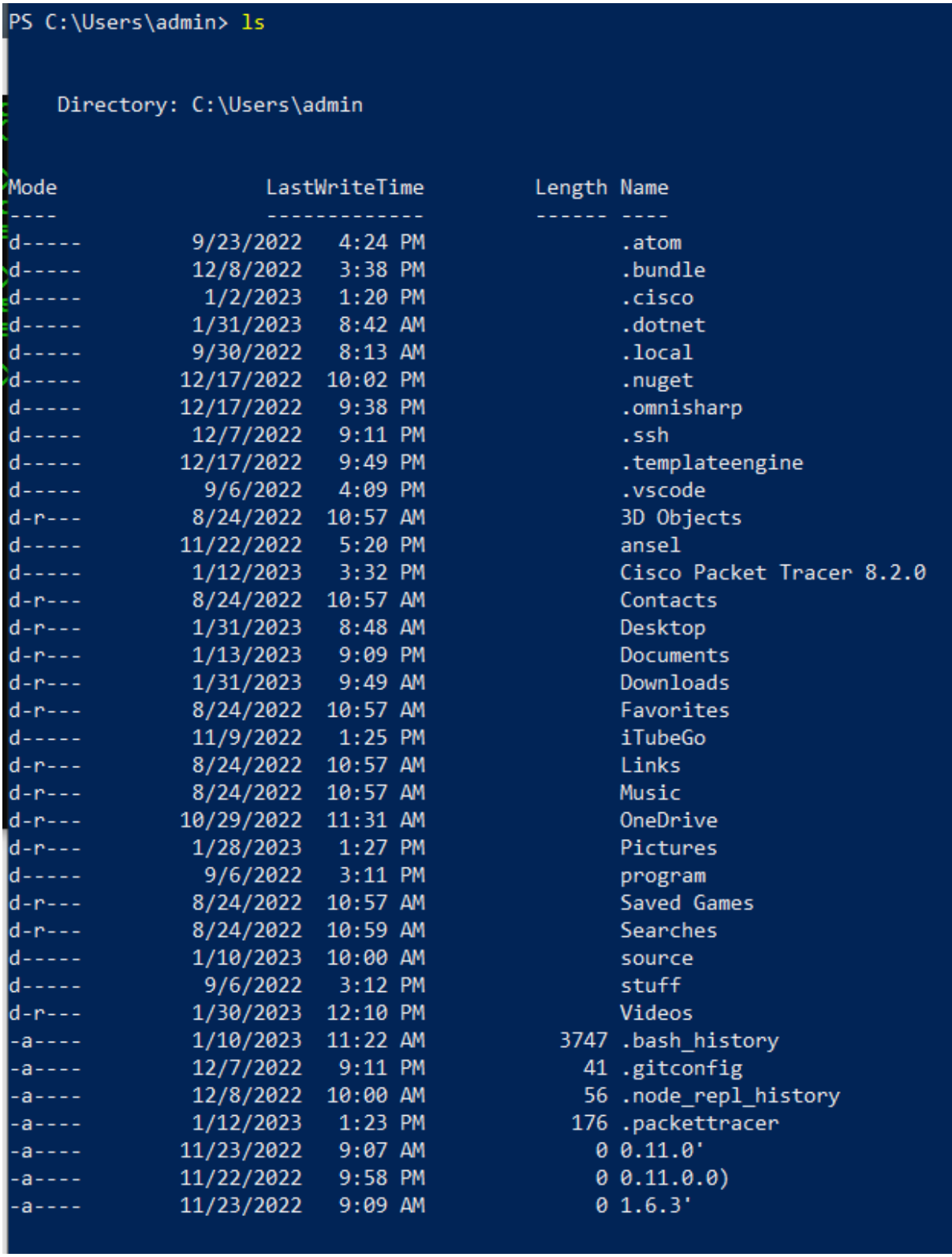
1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:
    a. cd: change directory



    b. ls: list out the content

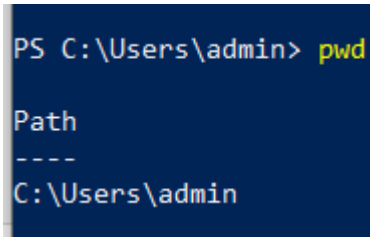


```
PS C:\Users\admin> ls


    Directory: C:\Users\admin


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/23/2022   4:24 PM                .atom
d-----         12/8/2022   3:38 PM                .bundle
d-----          1/2/2023   1:20 PM                .cisco
d-----         1/31/2023   8:42 AM                .dotnet
d-----         9/30/2022   8:13 AM                .local
d-----        12/17/2022  10:02 PM                .nuget
d-----        12/17/2022   9:38 PM                .omnisharp
d-----         12/7/2022   9:11 PM                .ssh
d-----        12/17/2022   9:49 PM                .templateengine
d-----          9/6/2022   4:09 PM                .vscode
d-r---         8/24/2022  10:57 AM                3D Objects
d-----        11/22/2022   5:20 PM                ansel
d-----         1/12/2023   3:32 PM                Cisco Packet Tracer 8.2.0
d-r---         8/24/2022  10:57 AM                Contacts
d-r---         1/31/2023   8:48 AM                Desktop
d-r---         1/13/2023   9:09 PM                Documents
d-r---         1/31/2023   9:49 AM                Downloads
d-r---         8/24/2022  10:57 AM                Favorites
d-----         11/9/2022   1:25 PM                iTubeGo
d-r---         8/24/2022  10:57 AM                Links
d-r---         8/24/2022  10:57 AM                Music
d-r---        10/29/2022  11:31 AM                OneDrive
d-r---         1/28/2023   1:27 PM                Pictures
d-----          9/6/2022   3:11 PM                program
d-r---         8/24/2022  10:57 AM                Saved Games
d-r---         8/24/2022  10:59 AM                Searches
d-----         1/10/2023  10:00 AM                source
d-----          9/6/2022   3:12 PM                stuff
d-r---         1/30/2023  12:10 PM                Videos
-a----         1/10/2023  11:22 AM           3747 .bash_history
-a----         12/7/2022   9:11 PM             41 .gitconfig
-a----         12/8/2022  10:00 AM             56 .node_repl_history
-a----         1/12/2023   1:23 PM            176 .packettracer
-a----        11/23/2022   9:07 AM              0 0.11.0'
-a----        11/22/2022   9:58 PM              0 0.11.0.0)
-a----        11/23/2022   9:09 AM              0 1.6.3'
```

c. pwd: path with directory, writes the full pathname of the current working directory to the standard output.



2. Consider the following kinds of information, and suggest the most appropriate data type to store or represent each:

Information	**Suggested Data Type**
A person's name	string
A person's age in years	integer
A phone number	integer
A temperature in Celsius	float
The average age of a group of people	float
Whether a person has eaten lunch	bool

3. Aside from the examples already provided in question 2, come up with an example of information that could be stored as:

Data type	**Suggested Information**
String	Kha
Integer	9
Float	9.9
Boolean	true

4. Fill out the following table, evaluating the value of each expression and identifying the data type the value is most likely to be:

Expression	**Given**	**Value**	**Data Type**
`6`		6	int
`True`		true	bool

`a`	`a = 2.5`	2.5	float
`1 + 2 * 3`		7	int
`a and False`	`a = True`	false	bool
`a or False`	`a = True`	true	bool
`a + b`	`a = 1` `b = 2`	3	int
`2 * a`	`a = 3`	6	int
`a * 2 + b`	`a = 2.5` `b = 2`	7	int
`a + 2 * b`	`a = 2.5` `b = 2`	6.5	float

`(a + b) * c`	`a = 1`
	`b = 1`
	`c = 5`
		10	int

`"Fred" + " Smith"`		Fred Smith	string
`a + " Smith"`	`a = "Wilma"`	Wilma Smith	str

5. Using an example, explain the difference between **declaring** and **initialising** a variable.
   The difference between the two is:
   Declaration tells the compiler about the existence of an entity in the program and its location.
   Initialization is the process of assigning a value to the Variable.

6. Explain the term **parameter**. Write some code that demonstrates a simple of use of a parameter. You should show a procedure or function that uses a parameter, and how you would call that procedure or function.

A parameter is a named variable passed into a function.
name is a parameter.
This is a function because it return a value, namely the processed parameter 'name'
#note the code used in the demonstration is ruby code

```ruby
def print_silly_name(name)
    i=0
    puts(name + " is a")
    while i < 60
        print 'silly '
        i=i+1
    end
    puts 'name!'
end
```

7. Using an example, describe the term **scope**.
   Scope is a concept that refers to where values and functions can be accessed For example, a variable in a class has a 'local scope' accessible only in that particular class whereas a declared public variable has 'global scope', wider access range.

8. In any procedural language you like, write a function called Average, which accepts an array of integers and returns the average of those integers. Do not use any libraries for calculating the average. You must demonstrate appropriate use of parameters, returning and assigning values, and use of a loop. Note — just write the function at this point, we'll *use* it in the next task. You shouldn't have a complete program or even code that outputs anything yet at the end of this question.
   #note the code used in the demonstration is ruby code

```ruby
def average()
    numbers=Array.new
    i=1
    while i<11
        print "Enter integer #{i}: "
        i+=1
        number=gets.chomp.to_i()
        numbers<<number
    end
    averages = numbers.sum(0.0) /numbers.size
    puts "average: #{averages}"
end
```

<insert a screenshot of your code here>

9. In the same language, write the code you would need to call that function and print out the result.

```
14    average()
15
PS D:\COS10009\labs\lab1> ruby pro2.rb
Enter integer 1: 5
Enter integer 2: 5
Enter integer 3: 5
Enter integer 4: 5
Enter integer 5: 5
Enter integer 6: 5
Enter integer 7: 5
Enter integer 8: 5
Enter integer 9: 5
Enter integer 10: 5
average: 5.0
```

<insert a screenshot of your code here>

10. To the code from 9, add code to print the message “Double digits” if the average is above or equal to 10. Otherwise, print the message “Single digits”. Provide a screenshot of your program running.

```
if averages < 10
    puts "it is indeed single digit"
end

if averages >=10 && averages<100
    puts "it is indeed double digit"
end
```

<insert a screenshot of your code here>

<insert a screenshot of your whole program running here>

```ruby
def average()
    numbers=Array.new
    i=1
    while i<11
        print "Enter integer #{i}: "
        i+=1
        number=gets.chomp.to_i()
        numbers<<number
    end
    averages = numbers.sum(0.0) /numbers.size
    puts "average: #{averages}"
    if averages < 10
        puts "it is indeed single digit"
    end

    if averages >=10 && averages<100
        puts "it is indeed double digit"
    end

end
average()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\COS10009\labs\lab1> ruby pro2.rb
Enter integer 1: 11
Enter integer 2: 12
Enter integer 3: 13
Enter integer 4: 14
Enter integer 5: 15
Enter integer 6: 16
Enter integer 7: 17
Enter integer 8: 18
Enter integer 9: 19
Enter integer 10: 20
average: 15.5
it is indeed double digit
```