

KARANAM YESWANTH TEJA

289224

BASH PROJECT

1) Bash Case

1.1)

```
karanamteja@334a7a24d06950c:~$ vi b1.sh
karanamteja@334a7a24d06950c:~$ chmod +x b1.sh
karanamteja@334a7a24d06950c:~$ ./b1.sh
Do you know Java Programming?
Yes/No? :yes
That's amazing.

karanamteja@334a7a24d06950c:~$ ./b1.sh
Do you know Java Programming?
Yes/No? :no
It's easy. Let's start learning from javatpoint.
karanamteja@334a7a24d06950c:~$ cat b1.sh
echo "Do you know Java Programming?"
read -p "Yes/No? :" Answer
case $Answer in
    Yes|yes|y|Y)
        echo "That's amazing."
        echo
        ;;
    No|no|N|n)
        echo "It's ea
sy. Let's start learning from javatpoint."
        ;;
    *)
        ;;
esac
```

1.2)

```
karanamteja@334a7a24d06950c:~$ nano b2.sh
karanamteja@334a7a24d06950c:~$ chmod +x b2.sh
karanamteja@334a7a24d06950c:~$ ./b2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name: windows
That's common. You should try something new.

karanamteja@334a7a24d06950c:~$ ./b2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name: android
This is my favorite. It has lots of applications.

karanamteja@334a7a24d06950c:~$ ./b2.sh
Which Operating System are you using?
Windows, Android, Chrome, Linux, Others?
Type your OS Name: others
Sounds interesting. I will try that.

karanamteja@334a7a24d06950c:~$ cat b2.sh
#!/bin/bash

echo "Which Operating System are you using?"
echo "Windows, Android, Chrome, Linux, Others?"
read -p "Type your OS Name: " OS

case $OS in
    Windows|windows)
        echo "That's common. You should try something new."
        echo
        ;;
    Android|android)
        echo "This is my favorite. It has lots of applications."
        echo
        ;;
    Chrome|chrome)
        echo "Cool!!! It's for pro users. Amazing Choice."
        echo
```

```
→ Chrome|chrome)
    echo "Cool!!! It's for pro users. Amazing Choice."
    echo
    ;;
Linux|linux)
    echo "You might be serious about security!!"
    echo
    ;;
*)
    echo "Sounds interesting. I will try that."
    echo
    ;;
esac

karanamteja@334a7a24d06950c:~$ _
```

2) For loop

2.1)

```
karanamteja@334a7a24d06950c:~$ nano b3.sh
karanamteja@334a7a24d06950c:~$ chmod +x b3.sh
karanamteja@334a7a24d06950c:~$ ./b3.sh
Start
learning
from
Javatpoint.
Thank You.
karanamteja@334a7a24d06950c:~$ cat b3.sh
#!/bin/bash
# This is the basic example of 'for loop'.

learn="Start learning from Javatpoint."

for learn in $learn
do
    echo $learn
done

echo "Thank You."
```

2.2) For loop to read a range

```
karanamteja@334a7a24d06950c:~$ nano b4.sh
karanamteja@334a7a24d06950c:~$ chmod +x b4.sh
karanamteja@334a7a24d06950c:~$ ./b4.sh
1
2
3
4
5
6
7
8
9
10
Series of numbers from 1 to 10.
karanamteja@334a7a24d06950c:~$ cat b4.sh
#!/bin/bash
# This is the basic example to print a series of numbers from 1 to 10.

for num in {1..10}
do
    echo $num
done

echo "Series of numbers from 1 to 10."
```

2.3) For Increment

```
karanamteja@334a7a24d06950c:~$ vi b5.sh
karanamteja@334a7a24d06950c:~$ chmod +x b5.sh
karanamteja@334a7a24d06950c:~$ ./b5.sh
1
2
3
4
5
6
7
8
9
10
karanamteja@334a7a24d06950c:~$ ./b5.sh
1
2
3
4
5
6
7
8
9
10
karanamteja@334a7a24d06950c:~$ cat b5.sh

for num in {1..10..1}
do
    echo $num
done
```

2.4) For Decrement

```
karanamteja@334a7a24d06950c:~$ nano b6.sh
karanamteja@334a7a24d06950c:~$ chmod +x b6.sh
karanamteja@334a7a24d06950c:~$ ./b6.sh
10
9
8
7
6
5
4
3
2
1
0
karanamteja@334a7a24d06950c:~$ cat b6.sh
for num in {10..0..1}
do
    echo $num
done
```

2.5) For loop to read array variables

```
karanamteja@334a7a24d06950c:~$ nano b7.sh
karanamteja@334a7a24d06950c:~$ chmod +x b7.sh
karanamteja@334a7a24d06950c:~$ ./b7.sh
WelcometoJavatpoint
karanamteja@334a7a24d06950c:~$ cat b7.sh
arr=( "Welcome""to""Javatpoint" )

for i in "${arr[@]}"
do
    echo $i
done
```

3) While loop

3.1) While Loop with single condition

```
karanamteja@334a7a24d06950c:~$ nano bw1.sh
karanamteja@334a7a24d06950c:~$ chmod +x bw1.sh
karanamteja@334a7a24d06950c:~$ ./bw1.sh
Enter starting number: 10
Enter ending number: 15
10
11
12
13
14
15
This is the sequence that you wanted.
karanamteja@334a7a24d06950c:~$ cat bw1.sh
#!/bin/bash
#Script to get specified numbers

read -p "Enter starting number: " snum
read -p "Enter ending number: " enum

while [[ $snum -le $enum ]];
do
    echo $snum
    ((snum++))
done

echo "This is the sequence that you wanted."
karanamteja@334a7a24d06950c:~$ _
```

3.2) While Loop with multiple conditions.

```
karanamteja@334a7a24d06950c:~$ nano bw2.sh
karanamteja@334a7a24d06950c:~$ chmod +x bw2.sh
karanamteja@334a7a24d06950c:~$ ./bw2.sh
Enter starting number: 10
Enter ending number: 17
10
11
12
13
14
15
16
17
This is the sequence that you wanted.
karanamteja@334a7a24d06950c:~$ cat bw2.sh
#!/bin/bash

read -p "Enter starting number: " snum
read -p "Enter ending number: " enum

while [[ $snum -lt $enum || $snum == $enum ]];
do
    echo $snum
    ((snum++))
done

echo "This is the sequence that you wanted."
```

3.3) Infinite while loop

```
karanamteja@334a7a24d06950c:~$ nano bw3.sh
karanamteja@334a7a24d06950c:~$ chmod +x bw3.sh
karanamteja@334a7a24d06950c:~$ cat bw3.sh
#!/bin/bash
```

```
while :
do
    echo "Welcome to Javatpoint."
done
```

```
karanamteja@334a7a24d06950c:~$ ./bw3.sh
```

[illegible]

3.4) While loop with break statement

```
karanamteja@334a7a24d06950c:~$ nano bw4.sh
karanamteja@334a7a24d06950c:~$ chmod +x bw4.sh
karanamteja@334a7a24d06950c:~$ ./bw4.sh
Countdown for Website Launching...
10
9
8
7
6
5
4
3
Mission Aborted, Some Technical Error Found.
karanamteja@334a7a24d06950c:~$ cat bw4.sh

echo "Countdown for Website Launching..."
i=10
while [ $i -ge 1 ]
do
    if [ $i == 2 ]
    then
        echo "Mission Aborted, Some Technical Error Found."
        break
    fi
    echo "$i"
    (( i-- ))
done

karanamteja@334a7a24d06950c:~$ _
```

3.5) While loop with continue statement

```
karanamteja@334a7a24d06950c:~$ nano bw5.sh
karanamteja@334a7a24d06950c:~$ chmod +x bw5.sh
karanamteja@334a7a24d06950c:~$ ./bw4.sh
Countdown for Website Launching...
10
9
8
7
6
5
4
3
Mission Aborted, Some Technical Error Found.
karanamteja@334a7a24d06950c:~$ ./bw5.sh
Current Number : 1
Current Number : 2
Current Number : 3
Current Number : 4
Current Number : 6
Current Number : 7
Current Number : 8
Current Number : 9
Current Number : 10
Current Number : 11
Skipped number 5 using Continue Statement.
karanamteja@334a7a24d06950c:~$ cat bw5.sh

i=0
while [ $i -le 10 ]
do
    ((i++))
    if [[ "$i" == 5 ]];
    then
        continue
    fi
    echo "Current Number : $i"
done

echo "Skipped number 5 using Continue Statement."
```

3.6) While Loop with C-style

```
karanamteja@334a7a24d06950c:~$ nano bw6.sh
karanamteja@334a7a24d06950c:~$ chmod +x bw6.sh
karanamteja@334a7a24d06950c:~$ ./bw6.sh
1
2
3
4
5
6
7
8
9
10
karanamteja@334a7a24d06950c:~$ cat bw6.sh
#!/bin/bash
#While loop example in C style

i=1
while((i <= 10))
do
    echo $i
    let i++
done
```

4) BASH UNTIL LOOP

4.1) Until loop with single condition

```
karanamteja@334a7a24d06950c:~$ nano bu1.sh
karanamteja@334a7a24d06950c:~$ chmod +x bu1.sh
karanamteja@334a7a24d06950c:~$ ./bu1.sh
1
2
3
4
5
6
7
8
9
10
karanamteja@334a7a24d06950c:~$ cat bu1.sh
i=1
until [ $i -gt 10 ]
do
echo $i
((i++))
done
```

4.2) Until loop with multiple condition

```
karanamteja@334a7a24d06950c:~$ nano bu2.sh
karanamteja@334a7a24d06950c:~$ chmod +x bu2.sh
karanamteja@334a7a24d06950c:~$ ./bu2.sh
a = 1 & b = 0.
a = 2 & b = 1.
a = 3 & b = 2.
a = 4 & b = 3.
a = 5 & b = 4.
karanamteja@334a7a24d06950c:~$ cat bu2.sh
max=5
a=1
b=0

until [[ $a -gt $max || $b -gt $max ]];
do
echo "a = $a & b = $b."
((a++))
((b++))
done
```

5) Bash String

5.1) Equal operator

```
karanamteja@334a7a24d06950c:~$ nano bs2.sh
karanamteja@334a7a24d06950c:~$ chmod +x bs2.sh
karanamteja@334a7a24d06950c:~$ ./bs2.sh
Strings are not equal.
karanamteja@334a7a24d06950c:~$ cat bs2.sh
str1="WelcometoJavatpoint."
str2="javatpoint"

if [ $str1 = $str2 ];
then
echo "Both the strings are equal."
else
echo "Strings are not equal."
fi
```

5.2) Not equal operator

```
karanamteja@334a7a24d06950c:~$ nano bs1.sh
karanamteja@334a7a24d06950c:~$ chmod +x bs1.sh
karanamteja@334a7a24d06950c:~$ ./bs1.sh
Strings are not equal.
karanamteja@334a7a24d06950c:~$ cat bs1.sh
str1="WelcometoJavatpoint."
str2="javatpoint"

if [[ $str1 != $str2 ]];
then
echo "Strings are not equal."
else
echo "Strings are equal."
fi
karanamteja@334a7a24d06950c:~$
```

5.3) Less than operator

```
karanamteja@334a7a24d06950c:~$ nano bs3.sh
karanamteja@334a7a24d06950c:~$ chmod +x bs3.sh
karanamteja@334a7a24d06950c:~$ ./bs3.sh
WelcometoJavatpoint is not less then Javatpoint
karanamteja@334a7a24d06950c:~$ cat bs3.sh
str1="WelcometoJavatpoint"
str2="Javatpoint"
if [ $str1 \< $str2 ];
then
    echo "$str1 is less then $str2"
else
    echo "$str1 is not less then $str2"
fi
```

5.4) Greater than operator

```
karanamteja@334a7a24d06950c:~$ nano bs4.sh
karanamteja@334a7a24d06950c:~$ chmod +x bs4.sh
karanamteja@334a7a24d06950c:~$ ./bs4.sh
WelcometoJavatpoint is greater then Javatpoint
karanamteja@334a7a24d06950c:~$ cat bs4.sh
str1="WelcometoJavatpoint"
str2="Javatpoint"
if [ $str1 \> $str2 ];
then
    echo "$str1 is greater then $str2"
else
    echo "$str1 is less then $str2"
fi
```

5.5) To check if the string length is greater than zero

```
karanamteja@334a7a24d06950c:~$ nano bs5.sh
karanamteja@334a7a24d06950c:~$ chmod +x bs5.sh
karanamteja@334a7a24d06950c:~$ ./bs5.sh
String is not empty
karanamteja@334a7a24d06950c:~$ cat bs5.sh
str="WelcometoJavatpoint"

if [ -n $str ];
then
    echo "String is not empty"
else
    echo "String is empty"
fi
```

5.6) To check if the string length is equal to zero

```
karanamteja@334a7a24d06950c:~$ nano bs6.sh
karanamteja@334a7a24d06950c:~$ chmod +x bs6.sh
karanamteja@334a7a24d06950c:~$ ./bs6.sh
String is empty.
karanamteja@334a7a24d06950c:~$ cat bs6.sh
str=""

if [ -z $str ];
then
    echo "String is empty."
else
    echo "String is non-empty."
fi
```

6) Bash Find

6.1) Find string length in bash - \${#str}

```
karanamteja@334a7a24d06950c:~$ nano bf.sh
karanamteja@334a7a24d06950c:~$ chmod +x bf.sh
karanamteja@334a7a24d06950c:~$ ./bf.sh
Length of 'Welcome to Javatpoint' is 21
karanamteja@334a7a24d06950c:~$ cat bf.sh
str="Welcome to Javatpoint"
length=${#str}

echo "Length of '$str' is $length"
karanamteja@334a7a24d06950c:~$
```

6.2) find string length in bash – expr

```
karanamteja@334a7a24d06950c:~$ nano bf1.sh
karanamteja@334a7a24d06950c:~$ chmod +x bf1.sh
karanamteja@334a7a24d06950c:~$ ./bf.sh
Length of 'Welcome to Javatpoint' is 21
karanamteja@334a7a24d06950c:~$ ./bf1.sh
Length of 'Welcome to Javatpoint' is 21
karanamteja@334a7a24d06950c:~$ cat bf1.sh
str="Welcome to Javatpoint"
length=`expr length "$str"`

echo "Length of '$str' is $length"
karanamteja@334a7a24d06950c:~$
```

6.3) find string length in bash – expr

```
karanamteja@334a7a24d06950c:~$ nano bf2.sh
karanamteja@334a7a24d06950c:~$ chmod +x bf2.sh
karanamteja@334a7a24d06950c:~$ cat bf2.sh
str="Welcome to the world"
length=`expr "$str" : '.*'`

echo "Length of '$str' is $length"
karanamteja@334a7a24d06950c:~$ ./bf2.sh
Length of 'Welcome to the world' is 20
```


6.4) find string length in bash – wc

```
karanamteja@334a7a24d06950c:~$ nano bf3.sh
karanamteja@334a7a24d06950c:~$ chmod +x bf3.sh
karanamteja@334a7a24d06950c:~$ cat bf3.sh
str="Welcome to Javatpoint"
length=`echo $str | wc -c`

echo "Length of '$str' is $length"
karanamteja@334a7a24d06950c:~$ ./bf3.sh
Length of 'Welcome to Javatpoint' is 22
```

6.5) find string length in bash - awk

```
karanamteja@334a7a24d06950c:~$ nano bf4.sh
karanamteja@334a7a24d06950c:~$ chmod +x bf4.sh
karanamteja@334a7a24d06950c:~$ ./bf4.sh
Length of 'Welcome to India' is 16
karanamteja@334a7a24d06950c:~$ cat bf4.sh
str="Welcome to India"
length=`echo $str | awk '{print length}'`

echo "Length of '$str' is $length"
```

7) Bash Split String

7.1) By Space

```
karanamteja@334a7a24d06950c:~$ cat bss1.sh
read -p "Enter any string separated by space: " str #reading string value
IFS=" " #setting space as delimiter
read -ra ADDR <<<"$str" #reading str as an array as tokens separated by IFS
for i in "${ADDR[@]}"; #accessing each element of array
do
echo "$i"
done
karanamteja@334a7a24d06950c:~$ nano bss1.sh
karanamteja@334a7a24d06950c:~$ ./bss1.sh
Enter any string separated by space: we welcome you to ust
we
welcome
you
to
ust
```

7.2) By Symbol

```
karanamteja@334a7a24d06950c:~$ nano bss2.sh
karanamteja@334a7a24d06950c:~$ chmod +x bss2.sh
karanamteja@334a7a24d06950c:~$ ./bss2.sh
Enter any string separated by colon(:) we:welcome:you:to:join:ust.

we
welcome
you
to
join
ust.

karanamteja@334a7a24d06950c:~$ cat bss
bss1.sh bss2.sh
karanamteja@334a7a24d06950c:~$ cat bss2.sh
read -p "Enter any string separated by colon(:) " str #reading string value
readarray -d : -t strarr <<<"$str" #split a string based on the delimiter ':'
printf "\n"
#Print each value of Array with the help of loop
for (( n=0; n < ${#strarr[*]}; n++ ))
do
echo "${strarr[n]}"
done
```

7.3) by another string

```
karanamteja@334a7a24d06950c:~$ nano bss3.sh
karanamteja@334a7a24d06950c:~$ chmod +x bss3.sh
karanamteja@334a7a24d06950c:~$ ./bss3.sh
declare -a array=( [0]="We" [1]="Welcome" [2]="You" [3]="On" [4]="Javatpoint")
karanamteja@334a7a24d06950c:~$ cat bss3.sh
str="WeLearnWelcomeLearnYouLearnOnLearnJavatpoint"
delimiter=Learn
s=$str$delimiter
array=();
while [[ $s ]];

do
array+=( "${s%%"$delimiter"}*" );
s=${s#"${delimiter}"};
done;
declare -p array
karanamteja@334a7a24d06950c:~$
```

7.4) by trim command

```
karanamteja@334a7a24d06950c:~$ nano bss4.sh
karanamteja@334a7a24d06950c:~$ ./bss4.sh
We
welcome
you
on
javatpoint.
karanamteja@334a7a24d06950c:~$ cat bss4.sh
my_str="We;welcome;you;on;javatpoint."
my_arr=($(echo $my_str | tr ';' '\n'))
for i in "${my_arr[@]}"

do
echo $i
done
karanamteja@334a7a24d06950c:~$
```

8) Bash Substring

8.1) To extract till specific characters from starting

```
karanamteja@334a7a24d06950c:~$ nano bsubs1.sh
karanamteja@334a7a24d06950c:~$ chmod +x bsubs1.sh
karanamteja@334a7a24d06950c:~$ ./bsubs1.sh
String: We welcome you on ust
Total characters in a String: 21
Substring: We welcome
Total characters in Substring: 10
karanamteja@334a7a24d06950c:~$ cat bsubs1.sh
#!/bin/bash
#Script to extract first 10 characters of a string
echo "String: We welcome you on ust"
str="We welcome you on ust"
echo "Total characters in a String: ${#str} "
substr="${str:0:10}"
echo "Substring: $substr"
echo "Total characters in Substring: ${#substr} "
karanamteja@334a7a24d06950c:~$ _
```

8.2) To extract from specific characters onwards

```
karanamteja@334a7a24d06950c:~$ nano bsubs2.sh
karanamteja@334a7a24d06950c:~$ chmod +x bsubs2.sh
karanamteja@334a7a24d06950c:~$ ./bsubs2.sh
you on ust
karanamteja@334a7a24d06950c:~$ cat bsubs2.sh
#!/bin/bash
#Script to print from 11th character onwards
str="We welcome you on ust"
substr="${str:11}"
echo "$substr"
karanamteja@334a7a24d06950c:~$ _
```

8.3) To extract a single character

```
karanamteja@334a7a24d06950c:~$ nano bsubs3.sh
karanamteja@334a7a24d06950c:~$ chmod +x bsubs3.sh
karanamteja@334a7a24d06950c:~$ ./bsubs3.sh
y
karanamteja@334a7a24d06950c:~$ cat bsubs3.sh
#!/bin/bash
#Script to print 11th character of a String
str="We welcome you on ust"
substr="${str:11:1}"
echo "$substr"
```

8.4) To extract the specific characters from last

```
karanamteja@334a7a24d06950c:~$ nano bsubs4.sh
karanamteja@334a7a24d06950c:~$ chmod +x bsubs4.sh
karanamteja@334a7a24d06950c:~$ ./bsubs4.sh
you on ust.
karanamteja@334a7a24d06950c:~$ cat bsubs4.sh
#!/bin/bash
#Script to extract 11 characters from last
str="We welcome you on ust."
substr="${str:(-11)}"
echo "$substr"
```

9) Bash concatenate String

9.1) Write variables side by side

```
karanamteja@334a7a24d06950c:~$ nano bc1.sh
karanamteja@334a7a24d06950c:~$ chmod +x bc1.sh
karanamteja@334a7a24d06950c:~$ ./bc1.sh
We welcome you to UST.
karanamteja@334a7a24d06950c:~$ cat bc1.sh
#!/bin/bash
#Script to Concatenate Strings
#Declaring the first String
str1="We welcome you"
#Declaring the Second String
str2=" to UST."
#Combining first and second string
str3="$str1$str2"
#Printing a new string by combining both
echo $str3
karanamteja@334a7a24d06950c:~$
```

9.2) Using Double Quotes

```
karanamteja@334a7a24d06950c:~$ nano bc2.sh
karanamteja@334a7a24d06950c:~$ chmod +x bc2.sh
karanamteja@334a7a24d06950c:~$ ./bc2.sh
We welcome you to UST.
karanamteja@334a7a24d06950c:~$ cat bc2.sh
#!/bin/bash
#Script to Concatenate Strings
#Declaring String Variable
str="We welcome you"
#Add the variable within the string
echo "$str to UST."
```

9.3) Using append operator with loop

```
karanamteja@334a7a24d06950c:~$ nano bc3.sh
karanamteja@334a7a24d06950c:~$ chmod +x bc3.sh
karanamteja@334a7a24d06950c:~$ ./bc3.sh
Printing the name of the programming languages
javapythonCC++
karanamteja@334a7a24d06950c:~$ cat bc3.sh
#!/bin/bash
echo "Printing the name of the programming languages"
#Initializing the variable before combining
lang=""
#for loop for reading the list
for value in 'java''python''C''C++';
do
lang+="$value " #Combining the list values using append operator
done
#Printing the combined values
echo "$lang"
karanamteja@334a7a24d06950c:~$
```

9.4) Using the printf function

```
karanamteja@334a7a24d06950c:~$ nano bc4.sh
karanamteja@334a7a24d06950c:~$ chmod +x bc4.sh
karanamteja@334a7a24d06950c:~$ ./bc4.sh
Welcome to UST.
karanamteja@334a7a24d06950c:~$ cat bc4.sh
#!/bin/bash
str="Welcome"
printf -v new_str "$str to UST."
echo $new_str
karanamteja@334a7a24d06950c:~$
```

9.5) Using Literal Strings

```
karanamteja@334a7a24d06950c:~$ nano bc5.sh
karanamteja@334a7a24d06950c:~$ nano bc5.sh
karanamteja@334a7a24d06950c:~$ chmod +x bc5.sh
karanamteja@334a7a24d06950c:~$ ./bc5.sh
Welcome to UST.
karanamteja@334a7a24d06950c:~$ cat bc5.sh
#!/bin/bash
str="Welcome to"
newstr="${str} UST."
echo "$newstr"
karanamteja@334a7a24d06950c:~$
```

9.6) Using Underscore

```
karanamteja@334a7a24d06950c:~$ nano bc6.sh
karanamteja@334a7a24d06950c:~$ chmod +x bc6.sh
karanamteja@334a7a24d06950c:~$ ./bc6.sh
Hello_World!
karanamteja@334a7a24d06950c:~$ cat bc6.sh
#!/bin/bash
str1="Hello"
str2="World!"
echo "${str1}_${str2}"
```

9.7) Using any Character

```
karanamteja@334a7a24d06950c:~$ nano bc7.sh
karanamteja@334a7a24d06950c:~$ chmod +x bc7.sh
karanamteja@334a7a24d06950c:~$ ./bc7.sh
Enter First Name: teja
Enter State: AP
Enter Age: 22
Name, State, Age: teja,AP,22
karanamteja@334a7a24d06950c:~$ cat bc7.sh
#!/bin/bash
#String Concatenation by Character (,) with User Input
read -p "Enter First Name: " name
read -p "Enter State: " state
read -p "Enter Age: " age
combine="$name,$state,$age"
echo "Name, State, Age: $combine"
```


10) Bash Functions

10.1) Syntax

```
karanamteja@334a7a24d06950c:~$ nano bfun1.sh
karanamteja@334a7a24d06950c:~$ chmod +x bfun1.sh
karanamteja@334a7a24d06950c:~$ ./bfun1.sh
Welcome to UST.
karanamteja@334a7a24d06950c:~$ cat bfun1.sh
JTP () {
echo 'Welcome to UST.'
}
JTP
karanamteja@334a7a24d06950c:~$ _
```

10.2) Passing Arguments

```
karanamteja@334a7a24d06950c:~$ nano bfun2.sh
karanamteja@334a7a24d06950c:~$ ./bfun2.sh
We
welcome
you
to
UST.
karanamteja@334a7a24d06950c:~$ cat bfun2.sh
function_arguments()
{
echo $1
echo $2
echo $3
echo $4
echo $5
}
#Calling function_arguments
function_arguments "We" "welcome" "you" "to" "UST."
karanamteja@334a7a24d06950c:~$ _
```

10.3) Variable Scope

```
karanamteja@334a7a24d06950c:~$ nano bfun3.sh
karanamteja@334a7a24d06950c:~$ chmod +x bfun3.sh
karanamteja@334a7a24d06950c:~$ ./bfun3.sh
Before Executing the Function
v1 is A.
v2 is B.
Inside Function
v1 is C.
v2 is D.
After Executing the Function
v1 is A.
v2 is D.
karanamteja@334a7a24d06950c:~$ cat bfun3.sh
#!/bin/bash
v1='A'
v2='B'
my_var () {
local v1='C'
v2='D'
echo "Inside Function"
echo "v1 is $v1."
echo "v2 is $v2."
}
echo "Before Executing the Function"
echo "v1 is $v1."
echo "v2 is $v2."
my_var

echo "After Executing the Function"
echo "v1 is $v1."
echo "v2 is $v2."
```

10.4) Return Values

```
karanamteja@334a7a24d06950c:~$ nano bfun4.sh
karanamteja@334a7a24d06950c:~$ chmod +x bfun4.sh
karanamteja@334a7a24d06950c:~$ ./bfun4.sh
Hello User
Hello Reader
The previous function returned a value of 5
karanamteja@334a7a24d06950c:~$ cat bfun4.sh
print_it () {
    echo Hello $1
    return 5
}
print_it User
    print_it Reader
    echo The previous function returned a value of $?
karanamteja@334a7a24d06950c:~$ _
```

10.5) Overriding Commands

```
karanamteja@334a7a24d06950c:~$ nano bfun5.sh
karanamteja@334a7a24d06950c:~$ chmod +x bfun5.sh
karanamteja@334a7a24d06950c:~$ ./bfun5.sh
[01-30 04:23:16] : Welcome to UST.
karanamteja@334a7a24d06950c:~$ cat bfun5.sh
echo () {
    builtin echo -n `date +"[%m-%d %H:%M:%S]"` ": "
    builtin echo $1
}

echo "Welcome to UST."
karanamteja@334a7a24d06950c:~$ _
```

11) Bash Array

```
Karanamteja@334a7a24d06950c:~$ nano barr1.sh
Karanamteja@334a7a24d06950c:~$ ./barr1.sh
printing the element with index of 2
ust

Printing all elements using @
Welcome To ust

using loop to iterate through all words
Welcome
To
ust

printing the keys of an array
0 1 2

Finding array length
The array contains 3 elements

#Script to print all keys and values using loop through the array
The key value of element Welcome is 0
The key value of element To is 1
The key value of element ust is 2

Adding Elements to an Array - method 1
Welcome To ust trivandrum
Adding elements to an Array - method 2
Welcome To ust trivandrum Kerala India

Updating Array Element
Welcome To UST_Campus trivandrum Kerala India

Deleting an element from an array
Welcome UST_Campus trivandrum Kerala India
```

```
Deleting the entire array
```

```
Slicing Array elements
Python
HTML
CSS
```

```
karanamteja@334a7a24d06950c:~$ cat barr1.sh
#!/bin/bash
#Script to print an element of an array with an index of 2
#declaring the array
declare -a example_array=( "Welcome" "To" "ust" )
echo "printing the element with index of 2"
echo ${example_array[2]}
echo

echo "Printing all elements using @ "
echo "${example_array[@]}"
echo

echo "using loop to iterate through all words"
for i in "${example_array[@]}"; do echo "$i"; done
echo

echo " printing the keys of an array"
echo "${!example_array[@]}"
echo

echo "Finding array length"
echo "The array contains ${#example_array[@]} elements"
echo

echo "#Script to print all keys and values using loop through the array"
for i in "${!example_array[@]}"
do
echo The key value of element "${example_array[$i]}" is "$i"
done
echo

echo "Adding Elements to an Array - method 1"
example_array[3]="trivandrum"
#Printing all the elements
echo "${example_array[@]}"
```

```
echo "Adding Elements to an Array - method 1"
example_array[3]="trivandrum"
#Printing all the elements
echo "${example_array[@]}"

echo "Adding elements to an Array - method 2"
example_array+=( Kerala India )
#Printing all the elements
echo "${example_array[@]}"
echo

echo "Updating Array Element"
example_array[2]=UST_Campus
#Printig all the elements of the Array
echo ${example_array[@]}
echo

echo " Deleting an element from an array"
unset example_array[1]
#Printing all the elements after deletion
echo "${example_array[@]}"
echo

echo "Deleting the entire array"
#Deleting Entire Array
unset example_array
#Printing the Array Elements
echo ${!example_array[@]}
#Printing the keys
echo ${!example_array[@]}
echo

echo "Slicing Array elements"
ex_array=( "Java" "Python" "HTML" "CSS" "JavaScript" )
#Slicing the Array
sliced_array=("${ex_array[@]:1:3}")
#Applying for loop to iterate over each element in Array
for i in "${sliced_array[@]}"
do
```