

# Operating System Laboratory

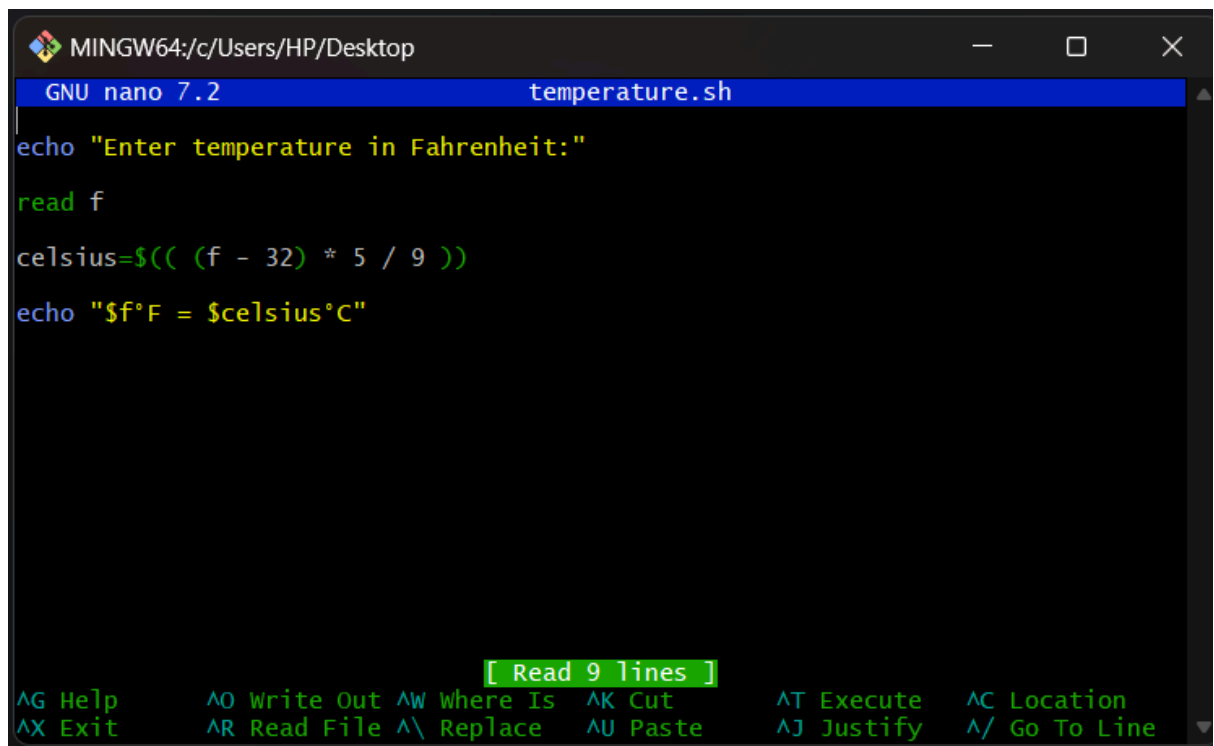
CSE-312

## Lab-3

### Question-1:

Convert Fahrenheit to Celsius

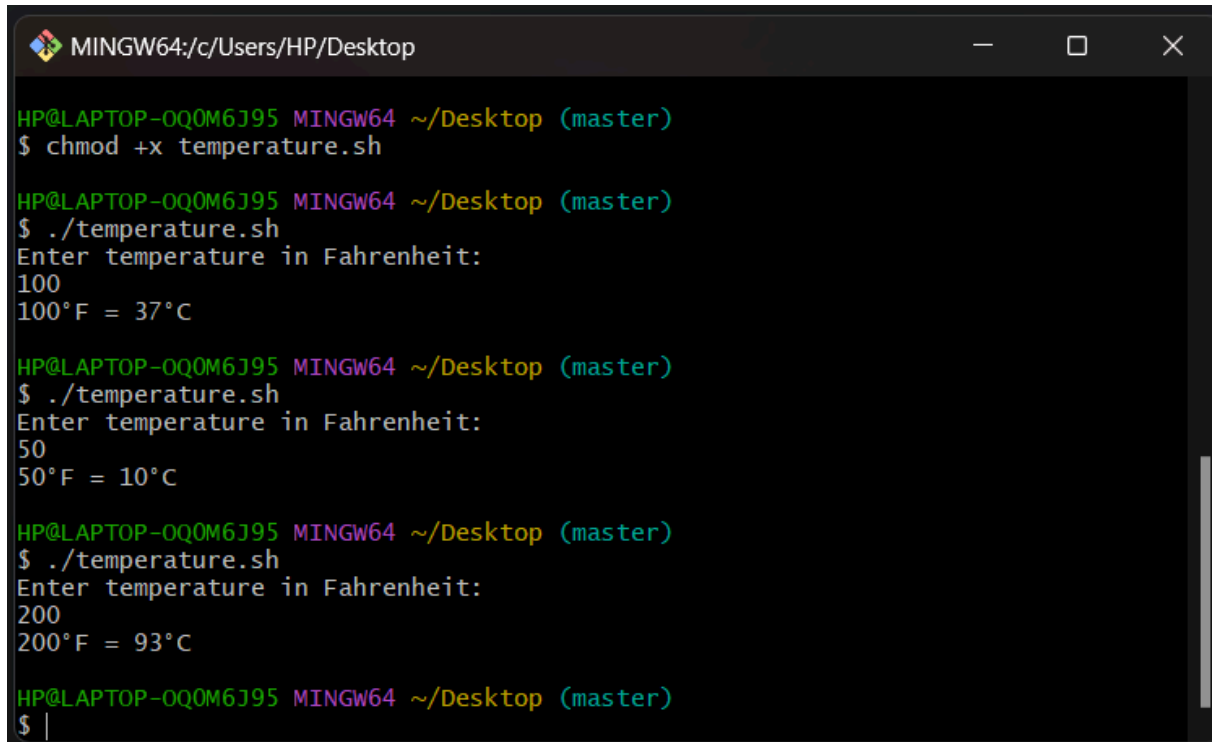
### Code:



```
MINGW64:/c/Users/HP/Desktop
GNU nano 7.2 temperature.sh
echo "Enter temperature in Fahrenheit:"
read f
celsius=$(( (f - 32) * 5 / 9 ))
echo "$f°F = $celsius°C"

[ Read 9 lines ]
^G Help    ^O Write Out ^W Where Is ^K Cut      ^T Execute  ^C Location
^X Exit    ^R Read File ^\ Replace  ^U Paste    ^J Justify  ^_ Go To Line
```

## Output:



```
MINGW64:/c/Users/HP/Desktop

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ chmod +x temperature.sh

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./temperature.sh
Enter temperature in Fahrenheit:
100
100°F = 37°C

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./temperature.sh
Enter temperature in Fahrenheit:
50
50°F = 10°C

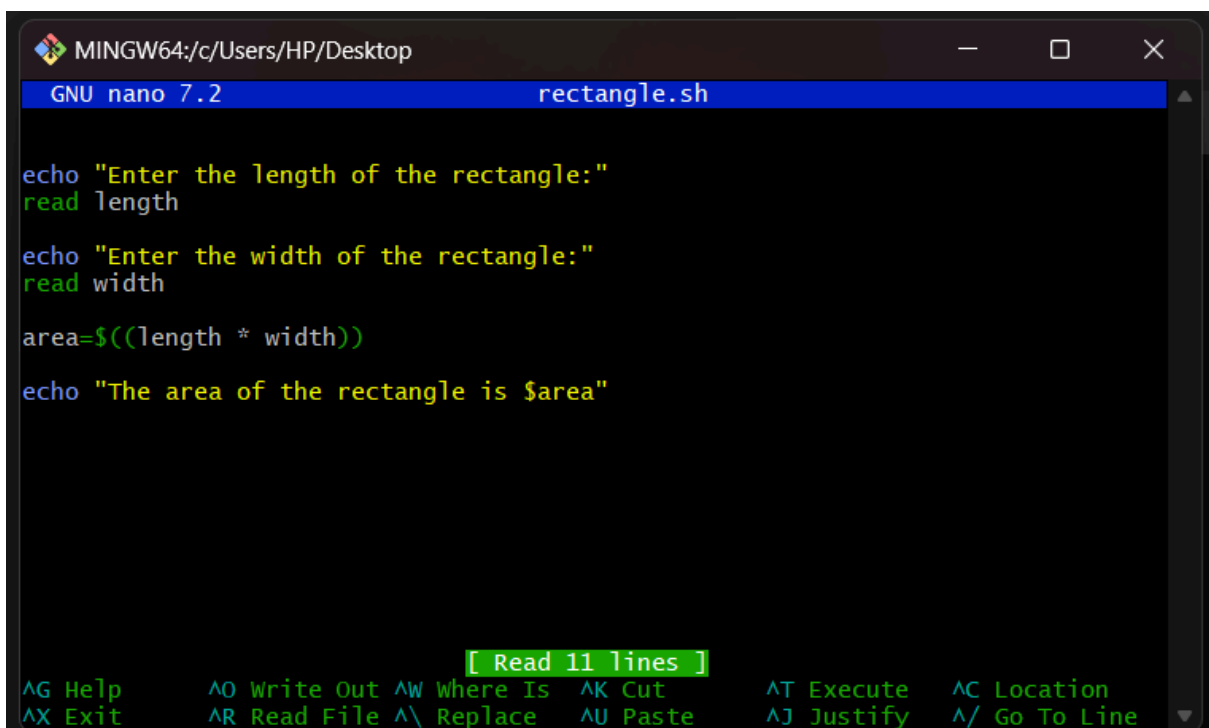
HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./temperature.sh
Enter temperature in Fahrenheit:
200
200°F = 93°C

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ |
```

## Question-2:

Calculate the Area of a Rectangle

## Code:



```
MINGW64:/c/Users/HP/Desktop
GNU nano 7.2 rectangle.sh

echo "Enter the length of the rectangle:"
read length

echo "Enter the width of the rectangle:"
read width

area=$((length * width))

echo "The area of the rectangle is $area"

[ Read 11 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

## Output:

```
MINGW64:/c/Users/HP/Desktop
HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ nano rectangle.sh

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ chmod +x rectangle.sh

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./rectangle.sh
Enter the length of the rectangle:
5
Enter the width of the rectangle:
6
The area of the rectangle is 30

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./rectangle.sh
Enter the length of the rectangle:
10
Enter the width of the rectangle:
10
The area of the rectangle is 100

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ |
```

## Question-3:

Calculate Grade with Number

## Code:

```
MINGW64:/c/Users/HP/Desktop
GNU nano 7.2 grade.sh

echo "Enter the student's score:"
read score

if [ $score -ge 80 ]
then
    echo "Grade: A+"
elif [ $score -ge 75 ]
then
    echo "Grade: A"
elif [ $score -ge 70 ]
then
    echo "Grade: A-"
elif [ $score -ge 65 ]
then
    echo "Grade: B+"
else
    echo "Grade: F"
fi

[ Read 19 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

## Output:

```
MINGW64:/c/Users/HP/Desktop

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ chmod +x grade.sh

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./grade.sh
Enter the student's score:
75
Grade: A

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./grade.sh
Enter the student's score:
90
Grade: A+

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./grade.sh
Enter the student's score:
35
Grade: F

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$
```

## Question-4:

Reverse a number

## Code:

```
MINGW64:/c/Users/HP/Desktop

GNU nano 7.2 reverse.sh Modified

echo "Enter a number:"
read num

reversed=0

while [ $num -gt 0 ]; do
    digit=$((num % 10))
    reversed=$((reversed * 10 + digit))
    num=$((num / 10))
done

echo "Reversed number: $reversed"

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

## Output:

```
MINGW64:/c/Users/HP/Desktop

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ chmod +x reverse.sh

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./reverse.sh
Enter a number:
34
Reversed number: 43

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./reverse.sh
Enter a number:
100
Reversed number: 1

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./reverse.sh
Enter a number:
969
Reversed number: 969

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$
```

## Question-5:

Perform a Logical Operation Based on User Input(AND,OR, NOT)

## Code:

```
MINGW64:/c/Users/HP/Desktop

GNU nano 7.2 logical.sh
echo "Enter a number"
read a
echo "Enter another number"
read b
if [ $a -eq 5 ] && [ $b -eq 5 ]
then
echo "And operation:True"
else
echo "And operation:False"
fi
if [ $a -eq 6 ] || [ $b -eq 6 ]
then
echo "Or operation:True"
else
echo "Or opearation:False"
fi
if [ $a -ne 7 ]
then
echo "Not operation:False"
else
echo "Not operation:True"
fi

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

## Output:

```
MINGW64:/c/Users/HP/Desktop
$ ./logical.sh
Enter a number
5
Enter another number
5
And operation:True
Or opearation:False
Not operation:False

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ chmod +x logical.sh

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$ ./logical.sh
Enter a number
7
Enter another number
8
And operation:False
Or opearation:False
Not operation:True

HP@LAPTOP-OQ0M6J95 MINGW64 ~/Desktop (master)
$
```