Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:	09	Week No:	09	Semester:	
Lecturer:	Name & email				

Lecture Outline

1. **Control Flow Statements**
2. Cases
3. Logical Operators

Control Flow Statements

❑ **if else statement**

  ❑ Syntax

  if [condition]

  then

       Statement

  else

       Statement

  fi

  *****There must be a space before and after the condition*****

       if [condition] not correct

    Must be written like this if [condition]

Control Flow Statements

- **if else if statement**

    - Syntax

        if [condition]; then

                Statement

        elif [condition]; then

                Statement

        else

                Statement

        fi

Example

```
count=10

if [ $count –eq 9 ]

then

        echo "Hello World"

else

        echo "Hello OS"

fi
```

Cases

❑   Used for multi level if conditions that means else if conditions.

❑   Syntax

   case expression in   [switch]

      pattern)   [case 1:]

            statement

      ;;   [break]

      pattern)   [case 2:]

            statement

      ;;   [break]

      *)   [default]

            echo Default

      ;;

   esac

Example

```
var="Rahim"
case $var in
        "Karim" )
                echo "Karim is good"
        ;;
        "Rahim" )
                echo "Rahim is bad"
        ;;
        * )
                echo "Unknown"
        ;;
esac
```

Logical Operators

❑ OR operator

```
val=10
if [ $val –gt 18 ] || [ $val –lt 30 ]; then
          statement
else
          statement
fi
```

❑ Another approach

```
val=10
if [ $val –gt 18 -o $val –lt 30 ]; then
          echo "statement 1"
else
          echo "statement"
fi
```

Logical Operators

❑ Another approach

```
val=10
if [[ $val –gt 18 -o $val –lt 30 ]]
then
        statement
else
        statement
fi
```

❑ AND operator

```
val=10
if [ $val –gt 18 ] && [ $val –lt 30 ]; then
        echo "$val is less then 30"
else
        echo "statement"
fi
```

Logical Operators

❑ Another approach
```
val=10
if [ $val –gt 18 -a $val –lt 30 ]
then
            statement
else
            statement
fi
```

❑ Another approach
```
val=10
if [[ $val –gt 18 -a $val –lt 30 ]]
then
            statement
else
            statement
fi
```

Books

❑ Unix Shell Programming

    ❑ Written by Yashavant P. Kanetkar