March 8th

→ Comments:-

):- // , / */ → Multi-line Comment.

↓
Single line Comment

(*) Input / output:-

=) output is basic operation performed to flow byte streams from main memory to the output device.

we can print any of the data types, ~~streams~~ patterns

functions used to print output to the console are:-

1) Print() → prints the message inside double Quotes

2) println() → prints the message inside double Quotes & moves to the begining of the next line.

println()

*):- Note:- print() → Internally calls System.out.println(), println()

(cmd+B)

Input:-

Kotlin standard input is basic operation performed to flow byte streams from input device such as keyboard to the main memory of system.

function used to take input:-

→ readline() → method

Scanner → class.

*Ex:-

```
var input = readline();
Print("$input");
```

Take input using scanner class:-

1) Step: 1) Import java.util.Scanner;

2) next.Int() for Integer I/P

next.Float() for Float I/P

next.Boolean() for boolean I/P

Q:- Take input from user with out scanner class.?

A):- code:-

```
val str = readline();
var Integer_value = string.toInt();   // Converting string to Integer
Print("your Integer_value is $Integer_value");
Print("enter double value");
val str1 = readline();
var double_value = str1.toDouble();   // String to double
Print("you entered : $double_value");
```

readline() function helps us take an input of only string data type.

=) ~~Type Conversion (Imp)~~:-

~~#~~ Basic input & output

→) code:-

```
fun main()
{
    val input = readline();
    Println("enter an Integer");
    var Integer_value = Input.toInt()
    println("The value of Integer is $Integer_value");
}
```

→ This program gives error at the line

→ var Integer_value = Input.toInt();

Reason:-

→ readline() returns nullable string, meaning it could be a null. In Kotlin, you cannot directly call methods on nullable types because it might result in "NullpointerException" at run time if the value is actually Null.

Q) How to resolve this issue then:-

A):- One way is to use safe call operator ?. and providing a default value if the input is null.

Code:-

```
val input = readline()
if( input != null)
    {  val Integer_value = input.toIntOrNull()
       if (IntegerValue != null)
         {  Println "The value of Integer is $Integer_value");
         }
       else{
            Print ("Invalid ");
           }
    }
    else {
            Print ("Invalid");
         }
```

-) toIntOrNull() attempts to convert the string to an Integer. It returns null if it is not valid representation of an Integer. This approach handles possibility of input being null and the input not being a valid Integer.

-) Using:- Scanner class:

```
Var Integer_value = Scanner( System.in );
Var value = Integer_value.nextInt();
```