compyler

Team Member:

- Muhammad Naim Siddique (EB 20102088)
- Muhammad Osama Ahmed (EB 20102092)
- Muhammad Sarim Khan (EB 20102096)
- Muhammad Umer Saleem (EB 20102098)
- Syed Huzaifa Ali (EB 20102128)

Documentation:

Programming Language: Compyler

Keywords:

Data Type	Logical Or
`str`	|
`int`	
`flt`	**Logical not**
`img`	`!`
`bool`	
`null`	**Logical And**
`Vect`	`&`
`Dict`	
`Set`	**Comparison operator**
	`Gt`
With Condition	`Lt`
`if`	`Gte`
`orif`	`Lte`
`until`	`Eq`
	`Neq`
Class	**Without Condition**
`Class`	`else`
	`try`
Function/Method	`atlast`
`prog`	
	Access Modifier
Import	`priv`
`use`	`pub`
	`prot`
Class Scope	**Assign**
`this`	`Is`

MDM	Comment
*	~ single line
/	`` multi line
%	
	SA
Operator	−
Incas (+=)	+
Decas (−=)	
Inc (++)	Expo
Dec (−−)	^
For	Paren
for	(
	)
Catch	{
catch	}
	[
	]

Code Example:

```
✓ prog main() return null {
      int number Is read("Enter the number: ");
      int total Is 0 ;
      int i;
      for i in [1 to number]{
          ~ adding all the numbers upto 'number'.
           total Incas i;
       }
      write("The total is ", total,".");
```

```
    }


  ✓ prog main() return null{
          int actual_number Is 12;
          bool flag Is True;
          until flag{
        `

        Continuously looping until the user guess the correct
        number.
        `

          int guess_number Is read("Guess the number :");
          if guess_number Eq actual_number{
              write("You won");
              flag Is False;
          }
          orif guess_number Gt actual_number{
              write("Too high");
          }
          else {
              write("Too low");
          }
      }
}
```