Variable Declaration

```
str customerName; // data type
must be declared
str custom erName = " Nam e";
const str consta ntV ariable =
" Val ue";
```

Primitive Data Types

`date`	dd/mm/yyyy
`enum`	named list of literals
`guid`	globally unique identifier
`int`	32-bit wide integer
`int64`	64-bit wide integer
`real`	number values with decimals
`str`	String
`timeOfDay`	number of seconds since midnight
`utcDat eTim e`	Combination of `date` and `timeOfDay`

Access Identifiers

[public]	Can be called from anywhere
[prote-cted]	called from methods in class/-subclass
[private]	only called in method where declared

Printing Infolog to Screen

```
info("This is an info log
message.");
```

Error Handling

```
throw error("This is an
error.");
```

Try...Catch

```
try
{
//Run some code
}
catch (Excep tio n:: Num eric)
{
info(' Found a Numeric except -
ion.');
}
catch
{
info(' Caught an except ion');
retry;
}
finally
{
            // Executed no
matter how the try block exits.
}
```

Assignment Operators

`=`	`int i = 1;`
`+=`	`int i; i=+1;`
`++`	`int i; i++;`
`-=`	`int i; i-=1;`
`--`	`int i; i--;`

Arithmetic Operators

`<<`	`int a = 1<<2; \\(1 x 2 x 2)`
`>>`	`int a = 1>>2; \\ (1/2/2)`
`*`	multiplies
`/`	divides
`div`	divides left value by right value
`mod`	divides with a remainder
`~`	performs a binary `NOT`operation
`&`	performs a binary `AND` operation
`^`	Binary `XOR` operation
`|`	binary `OR` operation
`+`	addition
`-`	subtraction
`?`	`int x = 4; result = (x ==5) ? ";`

Relational Operators

`Like`	SQL statement, returns criteria if true
`==`	`a==b\\ true`
`!=`	`a!=b \\true`
`>=`	`a>b`
`<=`	`a<=b`
`>`	`a>b`
`<`	`a<b`
`&&`	`a<b && b>c \\ AND`
`||`	`a>b || b>c \\ OR`
`!`	`a ! b \\ NOT`

If Statement

```
if (x == 5)
    {
            info('x equals
five');
    }
```

If...Else Statement

```
if (x == 5)
    {
info('x equals five');
    }
else
    {
info('x does not equal five');
    }
```

Switch Statement

```
switch (x){
case 5:
//do something
break;
case 10:
//do something
break;
default:
//do something
break;}
```

While Loop

```
int loopCount = 1;
container cont;
while (loopCount <= conlen -
(cont)) {
print conpee k(c ont ,lo opC -
ount);
loopCount = loopCount + 1;
}
```

For Loop

```
for (int i=1; i<=100; i+=1){
print ra[i];
}
```