GROUP No. 25

- MUHAMMAD USMAN (B21110006092)
- MUHAMMAD NAQEEBULLAH SHAH (B21110006084)

LANGUAGE SPECIFICATION

→ KEYWORD's :-

⇒ DATA TYPE's :-

- int
- float
- char
- str
- bool

→ DT

string

⇒ CONDITIONS :-

- provide
- except provide
- except

if

else

⇒ loop's :-

- loop
- do
- iter
- break
- Continue

while

for

⇒ function :-

- def
- return

void

thorw

ye 2no add kardena

⇒ OOP Concept :-

- Class
- Interface
- extends

- Implements
- Access modifiers
  - * private
  - * public
  - * [redacted]
- New

ye add karn dena in mein

this
super
static
instanceOf

→ OPERATOR's :-

⇒ Arithmetic operators

+, -, /, *, %

⇒ Increament & decreament operators

++, --

⇒ Assignment operators

=, +=, -=, *=, /=, %=

⇒ Relational operators

==, !=, >, <, >=, <=

⇒ logical operators.

&&, ||, !

&&,||,

→ PUNCTUATOR's :-

; {} [] ()

→ SYNTEX :-

⇒ Comment :-

- [redacted]
- Multi line comment.

// for single line comment
/// for multiple line comment

⇒ Variable declaration & initialization :-

- DT <name> ;
- DT <name> = <value> ;
- DT <name 1>, <name 2>, ---, <name n> ;

- DT <name 1> = <value 1>, <name 2> = <value 2>, ... <name n> = <value n>;
- <name> = <value>;
- <name 1> = <name 2> = ... = <name n OR value>

⇒ CONDITION'S :-

- provide (condition) {---}
- provide (condition) {---} except {---}
- provide (condition 1) {---} except provide (condition 2) {---} ---- except provide (condition n) {---} except {---}.

ye condition kar dena if lese ke hisaab se

⇒ Loop's :-

- while (condition) {---}
- do {---} while (condition)
- for (decl & initial; condition; incr & decr) {---}.

⇒ Array's :-

- DT [ ] <name> = { value } [value]
- <name> [index] = value;
- <name> = <name> [index]

⇒ functions.

- DT def <name> (DT <name 1>, DT <name 2>, ...) {
  return <var_name>; } ← function defining
- <name> (<value>, ---);
- <var_name> = <name> (<value>--); ← function calling.

⇒ OOP concept :-

→ Class defining :

- <Access_modifier> Class <name> {---}

→ Interface defining :

- <Access-modifier> interface <name> {---}

→ Constructor defining :

- <Access-modifier> <name> (DT <name>, ---) {---}

→ Object defining :

<Class_name> <object_name> = new <Class_name> ( );

→ Inheritance :

: <Access_modifier> Class <class_name> extends <Class_name> {---}

" <Access_modifier> Class <Class_name> implements <interface_name>, ---
--- {---}