Lize of data-types Ee format Specifiers: > * De use the size of () function to determine the size of any variable. % llu -> long long unsigned Size of () Zu Dinary unaigned Operators: > int a = 20; int b = 30; a + b = 50; result a,b -) oberands operands Sperator + -> addition arith metic Expression = > assignment O Akithmetic: -> +,-,/, *, %

11 ** (1) dosignment Bentons: > =, +=, -=, /=, *=, */0 = --a = a+2 = S+2 = 7Shorthand dentors a + = 2Short aut Augmented Compaison / Relational: $\neg > (<) > = (<) = (<)$ Unary/Inc/Dec/Prefix/Postfin: > Ternary Sperator / Shorthand if - else spenter/ > (Condition) ? true Value: false Value; Syntax: Mathematics (Toggle) -> Transfermers -> AC -> DC Bit Manipulation / Bit Maskeing (Sasken Bosch Bituise Right Shift >>>

(V) Bituise Left Shift >> <</td>

VI) Bituise NOT -> ~ int 728 int \rightarrow 2000BIN BIN \rightarrow 0000 int Shift Oberations: Value a = 10 unit/step = 2 15 -> 1111 >> 2 8 Lit Refresentation 1 1011 dis cardel 10>>2=2RS 1 32+16+8+4 15<<2 10 << 2 = 400100001111 60) 01000111100 Truth Table operator Bituise Not: ~ 5 = 10 -> Theoretical Value $\sim 5 = -6 \rightarrow \text{Practial Value}$ [-6-10] = 100 $-6 \rightarrow abs(-6) = 6 \rightarrow 0110$ M(0s) 2's tompliment $\rightarrow 1001$ n = 5 Formula $\rightarrow v(n) = -n-1=-6$ n = (-500) - n - 1-(-500) - 1 = 499So, for Betwise Not: n-) (absvalue) -> BIN 1 e com t Theoretical Answer: = 213 complement - Double Negation = Negation of 2 Practical Value: ~n Looping Branching Decision Making Losping Statements Jump Statements Conditional Statements * Simple if only oralition tif else > 2 condition * while loop break * do - while loss * continue * if else if else franz

* Switch case * goto * for loop (*) enhance for loop for each loop C++ / Java * tunay operator * Nested if - else Nested Loops Even or Odd Number -> n C> Not Div By 3 Div Ly 3 [Condition Traide Condition] Branching of Conditions