Ordinal types

- These refer to types where the possible values come from a finite set of values
- Subranges: e.g. the characters from ‘a’ to ‘z’, the integers from 37 to 103
- Enumerated sets: e.g. values 1, 27, and 103
- Provide better type checking and readability by associating a specific type name with the designated set of values

Subranges

- These can be supported as ordinal types as long as the range is finite and there is a defined ordering between the end points (e.g. the ascii/unicode values for characters, the numeric values of integers)
- Support for integer subranges often linked to syntax for specifying portions of a string or array in the language

Enumerated types

- Allow the developer to specify a finite set of values that are valid for a particular data type, e.g. a SpeedLimit can be any one of 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120
- Languages may restrict what types of values can be included in an enumerated type (e.g. C only permits integers or character) or specify that all values within an enumerated type must come from the same type
- Might allow association of identifiers with values (similar to constants), e.g. for the Weekdays type declare Sunday=0, Monday=1, etc
- Have to address issues of overlap and compatibility between types, e.g. can I assign SpeedLimit or Weekday values to integer variables?