1. The grammar that would allow the specified sentence “They ran fast to the exit”, is grammar B from page 925 in our textbook.



New lexical entries

Pronouns : he, she

Verb: jumped, threw, flew, spoke

Adv: far, quickly, up, slowly

Prep: from, at, away, over

Article: a, an

Noun: building, arena, cat, ball, stage, bird,

Grammar A:

English: He jumped quickly up to the stage

Non English: a cat caught away over an arena

Grammar B:

English: She ran far from the building

Non English: He threw quickly from at ball

Grammar c:

English: the bird flew slowly over the corpse

Non English: a building spoke far from a ball

1. Understanding grammar and syntax for both c++ and English is fairly straight forward. Both have a set of rules that need to be followed in order for a sentence or statement to be valid. So once you learn the rules those aspects are fairly easy to understand. In English semantics can be a little tough to understand because of things like idioms and so things like that may require some explaining. It’s the same thing with c++, you could have a function called x. While this would be allowed by the compiler, meaning its grammatically correct, it has no meaning to anyone other than the person who coded it. So generally in c++ comments are used to explain the semantics of a program. Pragmatically speaking English sentences are generally easy to understand. Once again, with c++ comments may be needed to inform others of the context of your code. As far as lexical ambiguity English has a fair amount of words that are lexically ambiguous like can, fly, sharp. The intended meaning can generally be determined by the full sentence but not always. For example, in class we were given the example “I can fish”. In this sentence the word can is lexically ambiguous and we cant really tell the actual meaning of this sentence without further information. Since this sentence could mean that one has the ability to fish or one stores fish inside of cans. Overall its hard to define what it means to actually understand something in English I would say that to understand something is to fully grasp the message how the author of said message intended. In c++ understanding would be being able to see why a program works the way it does.
2. Examples of dates :

Tuesday, December 1st,2015 04/05/1996

Friday 13th 30/4/2011

2018/3/5 2009 December 5th

Saturday 2nd January 2nd, 1987

02/12/85 March 06

Grammar:

D-> Day / MonthNum / Year

D-> MonthNum / Day / Year

D-> weekday, Month Day, Year

D-> weekday Day

Day -> 0 – 31

MonthNum -> 1 - 12

Month -> January - December

Year-> 0 – inf

weekDay -> Monday – Sunday