1) **The name should tell the intent**
   Why it exists, what it does, and how to use it
   E.g. good names would be **employeePyamentInfo** Vs **ePay**
2) **Avoid confusing names**
   Using names that already imply something
   E.g. naming something **unix**, **testList** (even if it, not a list)
3) **Choose clear names**
   Say what you mean and mean what you say
   E.g. **deleteItems** over **bustThemDown**, **kill** over **whack**
4) **Use good distinctions avoid using number ant end**
   Use distinctions that make sense and thus don't just use numbers
   E.g. **list1**, **list2** instead - **productIds**, **productDetails** etc
   E.g. using **productIds** and **productDetails** - means the same and distinction is harder between these two variables.
5) **Use pronounceable names**
   Programming is a social activity
   E.g. don't use the variable name as **dobyymm** for **DateOFBirthInYearsMonths**
6) **Use searchable names**
   Don't name variable as **'e'**, **'z'**,**8**, etc use - **Event**, **Max_Students**, etc
7) **Don't encode types in names**
   Remember containers of variables can change
   E.g. **phoneString**, **paymentInt**, etc are bad names, payment can be made float in the future and thus the name also has to now change.
8) **Avoid prefix to names**
   E.g. **m_description** -> **member_description** (easier to understand)
   E.g. **IShapeFactory** to mean it is an interface, instead, use **ShapeFactory** and **ShapeFactoryImpl**
9) **Class names - nouns**
   Function names - verbs
   E.g. Class names - **student**, **car**, **employee**, etc
   E.g. function names - **postPaymen**, **deletePage**, etc
10) **Use name consistently**
    Pick one concept and stick to it.
    E.g. **controller** everywhere Vs **manager** and **controller** used interchangeably, **driver** and **controller** used in the same place.
11) **Don't use the same name two mean two different things**
    E.g. payInfo to represent the amount to pay and payInfo to also represent who to pay and bank info, best employeePaymentAmount, and **employeeBankDetails**.
12) **Use domain-specific names**
13) **Remember your code is going to be read by computer engineers, helps them give context quickly**

E.g. **accountVisitor** (indicating visitor pattern), **jobQueue** - (indicating a queue), **nameBuilder** (indicating a builder)

14) Avoid - too long names

E.g. **m_description** -> **member_description** (easier to understand)