1.python qanday til?

J:Python, dasturlash va kompyuter ilmi sohalarida foydalaniladigan oddiy va kuchli bir dasturlash tili.

2.complatsiya boladigan til bilan compliyatsiya bomeydigan tilini farqi nima?

J:'Complatsiya' (composition) bir dasturlashda obyekt tuzilishining bir xususiyati hisoblanadi.

Bu, obyektlar yaratish uchun boshqa obyektlar yoki qo'shimcha funksiyalar ishlatishni anglatadi.

Misol uchun, bir klasda boshqa klasslarni qo'shib qo'yish mumkin.

Bu, obyektlarning funktsional so'raladigan bo'limlarini ajratib chiqish imkonini beradi.

'Compliyatsiya' (compilation) esa dasturni dastur bajarishga o'rgatish jarayonini anglatadi.

Bu, dasturlash tillaridagi kodni bina etish, debug qilish, va o'zgartirishdan so'ng kompyuter

uchun aniq instruksiyalar vaqtincha, quyidagi ko'rinishda, dastur bajarish uchun tarjima qilish jarayonidir.

Aslida, 'complatsiya' va 'compliyatsiya' so'zlari farqli kontseptlarni belgilaydi: biri dasturlashda

obyektlarning tuzilishi va boshqarilishiga, ikkinchisi esa dasturning bajarilish jarayoniga oid."

3.list va tuple ni nima farqi bor?

J:tuple Python dasturlash tillarida ma'lumotlarni saqlash uchun ishlatiladigan

o'zgaruvchanlar (variables) turidir. Farqi quyidagicha:

List:

O'zgaruvchanlar to'plami, bu degani yuqorida ko'rsatilgan.

Elementlar o'zgartirilishi mumkin.

Indekslar bo'yicha elementlarga murojat mumkin.

Sintaksis: [item1, item2, ...]

Tuple:

O'zgaruvchanlar to'plami, lekin unga qo'shimcha xususiyat sifatida o'zgarishi mumkin emas.

Elementlar o'zgartirilishi mumkin emas. Ya'ni, tuple qabul qilinayotgan o'zgaruvchanlar to'plami

bo'ladi, lekin ular o'zgartirib bo'lmaydi.

Indekslar bo'yicha elementlarga murojat mumkin.

Sintaksis: (item1, item2, ...)

4.list va dict ni farqi nima?

J:List va dictionary (dict) Python dasturlash tilida ma'lumotlarni saqlash uchun ishlatiladigan

ikkita turli to'plamadir, lekin ularning foydalanish usullari va qo'llanish xususiyatlari farqli:

List:

O'zgaruvchanlar to'plami, tartiblangan holatda ma'lumotlarni saqlaydi.

Indekslar bo'yicha elementlarga murojat mumkin.

Elementlar qatorda joylashadi va tartibini o'zgartirish mumkin.

Sintaksis: [item1, item2, ...]

Dictionary (dict):

Kalitlar va qiymatlar to'plami. Bu, kalitlar indekslar sifatida ishlaydi va qiymatlar bilan bog'liq bo'lgan ma'lumotlar saqlanadi.

Kalitlar unikal bo'lishi kerak. Har bir kalit faqat bir marta mavjud bo'lishi kerak.

Elementlar o'zgaruvchanlar to'plamiga o'xshash tartibda saqlanmaydi; ular kalitlar asosida qayta-tartiblanadi.

Sintaksis: {key1: value1, key2: value2, ...}

5.anonim funksiya nima va oddiy funksiyadan qanday farqi bor?

J:Oddiy funksiya (Named Function):

Oddiy funksiyalar nomlarini o'zida saqlaydi.

Funksiya nomi bilan murojaat qilinadi.

"def" kalit so'zi bilan boshlanadi.

Funksiyalar ko'p marta chaqirilishi mumkin.

Oddiy funksiyalarni qayta ishlatish va test qilish osonroqdir.

Anonim funksiya (Anonymous Function or Lambda Function):

Anonim funksiyalar nomlarga ega emas, ular o'zgaruvchi sifatida ishlatiladi.

"lambda" kalit so'zi bilan boshlanadi.

Oddiy funksiyalarga nisbatan qisqa vaqtda yoziladi.

Yalniz bir qatorlik ifodaga ega bo'lishi kerak.

Anonim funksiyalarni oddiy ifodalarga joylashtirish oson.

Asosiy farq anonim funksiyalar o'zgaruvchilar sifatida ishlatilishi va qisqa vaqt

ichida yozilishi bo'lib, ularni qayta ishlatish uchun oddiy funksiyalardan farqli ravishda nomga ega emaslar.

6.map va filter farqi nima?

J:map():

map() funksiyasi, barcha ro'yxatning har bir elementiga bitta funksiya qo'llanish imkonini beradi.

Natijalar ro'yxat ko'rinishida qaytariladi.

Kichikroq kod yozishga yordam beradi.

map() funksiyasi ikkita argument qabul qiladi: birinchi argument sifatida amalni bajarish uchun funksiya,

ikkinchi argument sifatida amal qilinishi kerak bo'lgan ro'yxat.

filter():

filter() funksiyasi, ro'yxatning har bir elementini tekshirib, berilgan shartni qanoatlantiradigan

elementlarni saqlash imkonini beradi.

Shartni qanoatlantirmaydigan elementlar o'chirib tashlanadi.

Natijalar ro'yxat ko'rinishida qaytariladi.

filter() funksiyasi ikkita argument qabul qiladi: birinchi argument sifatida tekshirish funksiyasi,

ikkinchi argument sifatida tekshirilishi kerak bo'lgan ro'yxat.

7.class,obyekt va instins ni farqini tushuntiring?

J:Class (Klass):

"Class" yaratish, o'zimizning yangi obyektlar yaratish uchun asosiy qurilmadir.

"Class" tuzilishi bilan, obyektlarning xususiyatlari (attributes) va metodlari (methods)ni aniqlab beradi.

"Class" tuzilishi "def" kalit so'zi bilan boshlanadi.

Object (Obyekt):

"Object" yoki "instance" obyektlar "class" ning mustaqil nusxasi, ya'ni amaliy birlashmalar.

"Class" dan nusxa olish orqali yaratiladi.

"Class" dagi attributlarni o'zida saqlaydi.

Obyekt yaratish uchun "class" ning nomi orqasiga argumentlar jo'natiladi.

Instance (O'rnatma):

"Instance" (o'rnatma) deb nomlangan narsa, bir "class" dan olingan bitta obyekt.

"Class" ning bitta nusxasi.

"Object" va "Instance" so'zi o'zaro almashtiriladi va bir xil ma'noni ifodalaydi.

8.classda constructor nima?

J:Constructor, klass obyektlarining yaratilishi va ularning boshlang'ich holatini belgilash uchun

ishlatiladigan maxsus bir metoddir. Python tilida bu metod `\_\_init\_\_()` nomi bilan nomlanadi.

Obyekt yaratilganda avtomatik ravishda chaqiriladi va berilgan argumentlar orqali obyektning

xususiyatlari (attributes)ga boshlang'ich qiymatlar o'rnatiladi.

9.oop dagi 4 ta tamoyilini tushuntirin?

J: Bundaylik (Encapsulation): Bundaylik, obyektlar orqali ma'lumotlar va funktsiyalar birlashmasini ifodalaydi.

Bunda obyektlar o'zlariga xos xususiyatlar va funksiyalarga ega bo'lishi mumkin. Bundaylik yordamida obyektning ichki holati yashirin qilib saqlanadi va faqat obyektning interfeysi orqali murojaat qilinishi mumkin.

Ibora (Abstraction): Ibora, kodning oddiylik va uni qayta ishlashga imkon beradi. Bu, obyektlarning faqat maqsul funksiyalar va ma'lumotlarni o'z ichiga olishi bilan aniqlanadi, va qo'shimcha tafsilotlar yoki dastur kodining qaysi qismlariga ega emas.

Boshqaruv (Inheritance): Boshqaruv, klasslar orasida hibrid klasslar yaratishni ta'minlaydi. Bunda bitta klassning barcha xususiyatlari va funksiyalari boshqa klassga miras oladi. Bu, kodni qayta ishlash va kodni qismlarni takrorlashni kamaytiradi.

Polimorfizm (Polymorphism): Polimorfizm, bir funktsiyani turli klasslarga yozish va shu funktsiyani turli turlar ustida ishlatish imkonini beradi. Bunda shakl o'zgartirish va funksiya ishlashining mazmuni o'zgarishi mumkin bo'lgan har qanday obyektga qarab ishlatishni ifodalaydi.

10.MRO nima?

J: MRO (Method Resolution Order), Python tilidagi klasslardagi metodlarni qaysi tartibda izohlab chiqish kerakligini aniqlash uchun ishlatiladigan

algoritmdir. MRO Python tilidagi klasslarning boshqaruvini aniqlash uchun foydalaniladi.

11.apstraktsiiya va inkopulatsiya ni farqi ?

J: Apstraktsiya:

Apstraktsiya, obyektlar yoki klasslar orqali ma'lumotlar va funktsiyalar birlashmasini ifodalashdir. Asosiy maqsadi kodning qisqa va oddiy bo'lishi, qo'shimcha tafsilotlarni yashirish va dastur kodining qaysi qismlariga ega emasligini ta'minlashdir. Apstraktsiyaga o'zgacha dasturiy interfeyslar va abstrakt klasslar yordamida ega bo'lish mumkin. Bu, turli klasslarning bir qismini yoki turli klasslar orasidagi o'zaro aloqalarni boshqarishda juda foydalanuvchilardan foydalaniladi.

Inkapsulyatsiya (Encapsulation): Inkapsulyatsiya, obyektlarning ichki holatini yashirib saqlash va faqat obyektni interfeysi orqali murojaat qilish imkonini beradi. Asosiy maqsadi ma'lumotlarni (xususiyatlarni va metodlarni) ochiq bo'lmagan holda saqlash va ularni faqat obyektni ichki logikasi orqali boshqarishdir. Bu, dasturlashda kodni qayta ishlashni kamaytiradi va obyektlarni qo'llashni osonlashtiradi.

12.dekerator nima?

J: Dekoratorlar Python tilida funktsiyalarni o'zgartirishsiz yoki uzaytirishsiz qo'llash uchun ishlatiladigan qulay vositalardir. Dekoratorlar, biror boshqa funktsiyani o'z ichiga oladi va unga qo'shimcha funksiyalarni qo'shish orqali funksiya ishlashini o'zgartiradi.

13.class ichidagi static metod nima?

J:

Klass ichidagi static metod, klassga tegishli bo'lsa-da, obyektlarga bog'liq emas va obyektlarning o'z xususiyatlariga yoki metodlariga murojaat qilmaydi. Bu metodlar obyektning ma'lumotlarini ishlash uchun ishlatilmaydi; ularni chaqirish uchun obyekt yoki klass nomi kerak emas.

Python tilida, static metodlar @staticmethod dekoratorini ishlatib aniqlanadi. Static metodlar klass obyektini almashish uchun biror bosh parametr qabul qilmaydi.

14.meta class nima?

J: Meta klass, Python tilidagi klasslarning "tungi" klassidir. Ya'ni, bu klasslar boshqa klasslarni belgilash uchun ishlatiladi. Meta klasslar, klass obyektlarini yaratishda va modifikatsiyalashda foydalaniladi. Meta klasslar, `type` klassidan miras oladi va klass obyektlarini yaratish uchun ishlatiladi. Meta klasslar ko'rsatilgan klasslarning tavsifi bilan shug'ullanadi. Meta klasslar bilan Python dasturlash tilida klasslar odatda ko'rsatilgan `class` kalit so'zi orqali yaratiladi.

15.mutable va immutable farqi nima?

J: Mutable (O'zgartiriluvchan): Mutable obyektlar o'z ichidagi ma'lumotlarni o'zgartirish imkonini beradi. O'zgartiriluvchan obyektlar ro'yxatlar (lists), lug'atlar (dictionaries), va ro'yxatlar bilan o'xshash turdagi obyektlar bo'lishi mumkin.

Misol uchun: ro'yxatlar, lug'atlar.

Immutable (O'zgartirilmas):Immutable obyektlar o'z ichidagi ma'lumotlarni o'zgartira olmaydi.

O'zgartirilmas obyektlar sonlar, matnlar, tuple'lar va frozenset'lar kabi turdagi obyektlar bo'lishi mumkin. Misol uchun: sonlar, matnlar, tuple'lar.

16.SQL da bor lekin PYTHONDA yo'q data type lari qaysilar?

J: REAL: Haqiqiy (o'nlik) sonlar uchun ma'lumot turidir.

TEXT: Matnlar (satrlar) uchun ma'lumot turidir.

DATE: Sana va vaqt ma'lumotlari uchun ma'lumot turidir.

17.SQL da SERIAL nima?

J: SQLda "SERIAL" data type bir maydon (column) turi hisoblanadi, uni odatda integer qiymatlar ketma-ketligini saqlash uchun ishlatiladi. Bu tur SQL ma'lumotlar bazalarida avtomatik ravishda o'sadigan va har bir qo'shilgan qatordan boshlab o'sadi. Yangi qator qo'shilganda, SERIAL maydoni o'zgartiriladi va o'sib boradi. Bu, odatda PRIMARY KEY sifatida ishlatiladi va unikal bo'lgan qiymatlar o'zgaruvchan sifatida olishi lozim.

18.JOIN lar haqida?

J: JOIN operatori SQL-da ma'lumotlar bazalari orasida bog'lanishni bajarish uchun ishlatiladi. Ma'lumotlar bazasidagi ikki yoki undan ko'p jadval (table) orasidagi ma'lumotlarni birgalikda chiqarish uchun ishlatiladi

19.RELATIONSHIP lar haqida?

J: Munosabatlar (relations) SQL ma'lumotlar bazalarida obyektlar (jadval yoki maydonlar) o'rtasidagi bog'lanishni ifodalaydi. Munosabatlar ikkita obyekt (jadval yoki maydon) o'rtasidagi bog'lanishni belgilaydi. Quyidagi asosiy munosabat turkumlari mavjud:

One-to-One (Bir-birga):

Bir-birga munosabatda, birinchi obyektning har bir qatori faqat bitta ikkinchi obyektning qatori bilan bog'lanadi. Misol uchun, shaxslar va ularning pasportlari orasidagi munosabat bir-birga munosabatga misol bo'lishi mumkin.

One-to-Many (Birga-ko'p):

Birga-ko'p munosabatda, birinchi obyektning har bir qatori bir nechta ikkinchi obyektning qatorlari bilan bog'lanadi. Masalan, bir universitet va undagi talabalar orasidagi munosabat birga-ko'p munosabatga misol bo'lishi mumkin, ya'ni har bir universitetda bir nechta talabalar bo'lishi mumkin.

Many-to-Many (Ko'p-ga-ko'p):

Ko'p-ga-ko'p munosabatda, birinchi obyektning har bir qatori bir nechta ikkinchi obyektning qatorlari bilan bog'lanadi va ikkinchi obyektning ham har bir qatori bir nechta birinchi obyektning qatorlari bilan bog'lanadi. Misol uchun, talabalar va fandanlar (fanlar) orasidagi munosabat ko'p-ga-ko'p munosabatga misol bo'lishi mumkin, ya'ni har bir talaba bir nechta fanga va har bir fan bir nechta talabaga tegishli bo'lishi mumkin.

20.DATABASE NORMALISATION nima?

J: Ma'lumotlar bazalari normalizatsiya, ya'ni ma'lumotlar o'zlashtirish jarayonidir. Bu, ma'lumotlar bazasidagi ma'lumotlar tuzilishini (bog'lanishini) yaxshilash, tozalash va o'rnatilganlikni oshirish jarayonidir. Normalizatsiya maqsadi, ma'lumotlar bazasidagi tozalash va redundansiyaning (o'zlashtirishning) kamayishiga olib keladi, shuningdek, ma'lumotlar bazasidagi muammo va bozor yonidagi qiyinchiliklarni kamaytiradi.

Ma'lumotlar bazalari normalizatsiyasi odatda "normalizatsiya darajalari" deb ataladi, ular "1-daraja"dan "5-darajagacha" bo'lgan.