அலகு I

பாடம் 2

தரவு அருவமாக்கம்

S5PLYA

கற்றலின் நோக்கங்கள்

இந்தப் பாடத்தைக் கற்றபின், மாணவர்கள் கற்றுக்கொள்வது,

- அருவமாக்க தரவு கட்டமைப்பை புரிந்து கொள்ளுதல்.
- அருவமாக்க தரவு வகை
- கான்கீரிட் மற்றும் அருவமாக்கம் தரவு வகை செயல்படுத்தலில் உள்ள வேற்றுமைகள்.
- இணை (Pairs)-ன் பயன்பாடு
- கட்டுருவில் தரவு அருவமாக்கம்.

2.1 தரவு அருவமாக்கம் அறிமுகம்

தரவு அருவமாக்கம் கணினி அறிவியலில் சக்தி வாய்ந்த கருத்துருவாகும். இது நிரலர்களை, நிரல் குறியீட்டை ஒரு பொருளாக கருத வழிச் செய்கிறது. உதாரணமாக வாகனம், பென்சில் மக்கள் போன்றவற்றை போல் நிரலையும் ஒரு பொருளாகக் கருதலாம். நிரலர்கள் நிரல் எவ்வாறு செயல்படுத்தப்படுகின்றது என்பதைப் பற்றி கவலைப்பட வேண்டிய அவசியம் இல்லை, இது அவர்கள் அப்பொருள் என்ன செய்கின்றது என்று அறிந்திருந்தால் போதும்.

குறிப்பாக பல நிரலர்கள் ஈடுபடும் செயல்திட்டத்தை (Project) செயல்படுத்தும் போது இது முக்கியமானதாகும். இங்கு செயல் திட்டம் என்பது நிரலாக்கத்தை குறிக்கிறது. தரவு அருவமாக்கத்தில் குழு உறுப்பினர்கள், குறிமுறையை புரிந்து கொள்ள ஒவ்வொரு வரியாக வாசிக்க தேவையில்லை, அவர்கள் அது ஒரு பணியினை செய்கிறது என்று கருதி கொண்டால் போதும்.

அருவமாக்கம் நிரலுக்கு கூறுநிலைமையை வழங்குகிறது. (ஒரு நிரல் பல தொகுதிகளாக பிரிக்கப்படுவது கூறு நிலையாகும்) இனக்குழுக்கள் போன்றவை அருவமாக்க தரவு வகையின் ("Abstract Data Types,"(ADT)) உருவமைப்பாகும்.

2.2 தரவு அருவமாக்கின் வகைகள்

அருவமாக்க தரவு வகை (Abstract Data type (ADT)) என்பது பொருள்களுக்கான வகை (அல்லது இனக்குழு) ஆகும், இதன் செயல் மதிப்பின் தொகுப்பு மற்றும் செயல்பாடுகளின் தொகுப்பால் வரையறுக்கப்படுகிறது.

ADT யின் வரையறுப்பு என்ன வகையான செயல்பாடுகள் செய்யப்பட வேண்டும் என்று குறிப்பிடுகின்றது, அவை எப்படி செயல்பட வேண்டும் என்று குறிப்பிடப்படுவதில்லை. இவை தரவுகள் நினைவகத்தில் எவ்வாறு சேமிக்கப்படுகின்றது என்றோ அல்லது எந்த நெறிமுறைகள் செய்யப்பட வேண்டும் என்ற வழிமுறையோ குறிப்பிடப்படுவதில்லை. சுயமாக இவை செயல்படுத்தப்படுவதில்லை எனவே, இதை அருவமாக்கம் என்று அழைக்கிறோம். விவரங்களை மறைத்து அவசியமானவற்றை மட்டும் வழங்கும் செயல்முறையை அருவமாக்கம் என்கிறோம்.

இவ்வகை வரையறைகள், ADTs எவ்வாறு உருவமைக்கப்படுகின்றன என்றோ அல்லது எவ்வாறு செயல்பாடுகள் கையாளப்படுகிறதோ என்று குறிப்பிடப்படுவதில்லை என்பதை நீங்கள் காணலாம். ADT யை பல்வேறு வகையில் செயல்படுத்தலாம், உதாரணமாக List ADT ஒற்றை இணைக்கப்பட்ட பட்டியல் அல்லது இரட்டை இணைக்கப்பட்ட பட்டியல் கொண்டு

செயல்படுத்தப்படலாம். இதுபோன்றே அடுக்கு (Stack) ADT மற்றும் வரிசை ADT, List பயன்படுத்தி செயல்படுத்தப்படலாம்.

தரவு அருவமாக்கம் நாம் இந்த உலகை பற்றி நினைப்பதை பிரிதிப்பலிக்கிறது. உதாரணமாக, ஒருவர் மகிழுந்தை ஓட்டி செல்வதற்கு, அதன் பொறி இயந்திரம் எவ்வாறு கட்டமைக்கப்பட்டது என்றோ டயர்கள் எவ்வகையான பொருட்கள் கொண்டு தயாரிக்கப்பட்டுள்ளது என்றோ தெரிந்து வைத்திறுக்க வேண்டியதில்லை மகிழுந்து எவ்வாறு இயக்கப்பட வேண்டும் என்ற பயிற்சி இருந்தால் போதும், தரவு அருவமாக்கினை செயல்படுத்த, ஆக்கிகள் (Constructor) மற்றும் செலக்டர் (Selectors) என்ற இரண்டு செயற்கூறுகள் உருவாக்கப்பட வேண்டும்.

2.3 ஆக்கிகள் மற்றும் செலக்டர்கள் (constructors and selectors)

ஆக்கி செயற்கூறுகள் அருவமாக்கம் தரவு வகையை கட்டமைக்க பயன்படுகிறது. செலக்டர் செயற்கூறுகள் தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது.

எடுத்துக்காட்டாக, *city* என்று ஒரு அருவமாக்க தரவு வகை உள்ளது என வைத்துக் கொள். *City* என்ற பொருள் நகரத்தின் பெயர், அட்சரேகை மற்றும் தீர்க்கரேகை பற்றிய தகவல்களை சேமித்திருக்கும் *city* என்ற பொருளை உருவாக்க பின்வரும் செயற்கூற்றினை பயன்படுத்தலாம்.

city = makecity (name, lat, lon)

city பொருளின் தகவல்களை பெறுவதற்கு பின்வரும் செயற்கூறுகளை பயன்படுத்தலாம்.

- getname(city)
- getlat(city)
- getlon(city)

பின்வரும் போலி குறிமுறை, இரு நகரங்களுக்கு இடையேயான தொலைதுரத்தை கணக்கிடும்:

```
distance(city1, city2):
    lt1, lg1 := getlat(city1), getlon(city1)
    lt2, lg2 := getlat(city2), getlon(city2)
      return ((lt1 - lt2)**2 + (lg1 - lg2)**2))^(1/2)
```

மேலே காணும் குறிமுறையில், distance(),getlat() மற்றும் getlon() ஆகியவை செயற்கூறுகள் ஆகும். lt என்பது அட்ச ரேசை மற்றும் lg என்பது தீர்க்கரேகையும் குறிக்கிறது. longitude. := என்பதை “assigned as” அல்லது “becomes” என்று வாசிக்க வேண்டும்.

lt1, lg1 := getlat(city1), getlon(city1)

இதனை lt1 என்பது getlat(city1)ன் மதிப்பாகிறது மற்றும் lg1 என்பது getlont(city1) ன் மதிப்பாகிறது. என்று வாசிக்க வேண்டும்.

இந்த செயற்கூறுகள் எவ்வாறு செயல்படுத்தப்படுகிறது என்பதை தெரிந்து கொள்ள வேண்டியதில்லை. இதை வேறு ஒருவர் நமக்காக வரையறுத்துள்ளார் என்று கருதிக் கொள்ள வேண்டும். பயனர் செயற்கூறுகள் எப்படி செயல்படுத்தப்படுகின்றது என்பது தெரிந்து வைத்திருக்க வேண்டியதில்லை எனினும் வேறு ஒருவரால் இச்செயற்கூறுகள் வரையறுக்கப்பட்டிருக்க வேண்டும்.

மேலே காணும் குறிமுறையில், ஆக்கி மற்றும் செலக்டர்களை அடையாளம் காண்போம். கண்ஸ்டரக்டரஸ் செயற்கூறுகள் அருவமாக்க தரவு வகையை கட்டமைக்கப் பயன்படுகிறது என்பதை நாம் அறிவோம். மேலே காணும் போலிக் குறிமுறையில், cityயின் பொருளை உருவாக்கும் செயற்கூறு, ஆக்கி ஆகும்.

city = makecity(name, lat, lon)

இங்கு makecity(name, lat, lon) என்ற ஆக்கி city எனும் பொருளை உருவாக்குகிறது.

Chapter 2.indd 13 30-01-2020 19:08:01

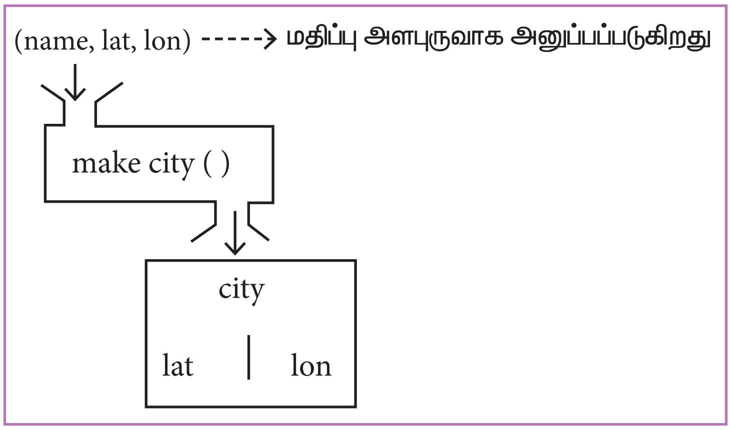
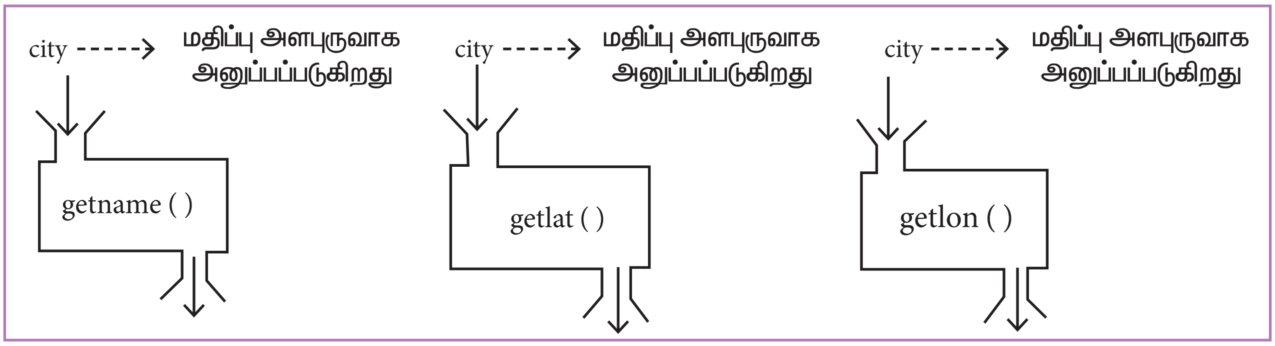


Fig 1 constructor

செலக்டார் செயற்கூறுகள் தகவல்களை தரவு வகையிலிருந்து பெறுவதற்கு பயன்படுகிறது. மேலே காணும் குறிமுறையில்,

- getname(city)
- getlat(city)
- getlon(city)

என்பவை city எனும் பொருளிலிருந்து தகவல்களை பெற்றுத் தரும் செலக்டர் செயற்கூறுகளாகும்.



படம் 2 செலக்டார்கள்

ஆக்கி மற்றும் செலக்டர்களை அடையாளம் காணும் மேலும் ஒரு எடுத்துக்காட்டைக் காணலாம். --என்ற குறியீட்டை குறிப்புரைகளாக என (Comments) என வாசிக்கவும்.

```
- - constructor
makepoint(x, y):
    return x, y
- - selector
 xcoord(point):
    return point[0]
- -selector
ycoord(point):
    return point[1]
```


Chapter 2.indd 14 30-01-2020 19:08:01

குறிப்பு

தரவு அருவமாக்கம் என்பது அருவமாக்க தரவு இனம் (Abstract Data Type (ADT)), அறிவிக்கப்பயன்படுகிறது. இது ஆக்கி மற்றும் செலக்டாரஸின் தொகுப்பாகும். ஆக்கிகள், பல்வேறு தகவல் துணுக்குகளை கொண்டு பொருள் உருவாக்கும், செலக்டார்ஸ் ஒவ்வொரு சிறு தகவல்களை பொருளிலிருந்து பெற உதவுகிறது.

2.4 விகிதமுறு எண்களைக் கொண்டு அருவமாக்க தரவு வகையை உருவமைப்பு செய்தல்

தரவு அருவமாக்கத்தின் அடிப்படை நோக்கம், நிரலினை அருவமாக்க தரவில் இயக்கும்படி கட்டமைப்பதாகும். அதாவது, நிரலில் தரவுகளை பயன்படுத்திக் கொள்ளும் போது அதைப்பற்றி சில ஊகங்களை செய்து கொள்ள வாய்ப்புள்ளது. அதை வேளையில் கான்கீரிட் தரவு உருவமைக்கும் போது நிரலின் தனிப்பகுதியாக வரையரை செய்ய வேண்டும்.

குறிப்பு

கான்கீரிட் தரவு உருவமைப்பில், அனைத்து செயற்கூறுகளின் வரையறையும் தெரிந்திருக்க வேண்டும்.

அனைத்து நிரல்களும் இரு பகுதிகளை கொண்டிருக்கும். அருவமாக்க தரவின் மீது செயல்படும் பகுதி மற்றும் அருவமாக்க தரவு வகையை செயல்படுத்தும் சில செயற்கூறுகளின் தொகுப்பை கொண்டு கான்கீரிட் உருவமைப்பை வரையறுக்கும் பகுதி என இரு பிரிவுகளை கொண்டிருக்கும். இந்த நுட்பத்தை புரிந்து கொள்வதற்கு, விகித முறு எண்களை கையாளும் சில செயற்கூறுகளுடன் வடிவமைக்கப்பட்ட எடுத்துக்காட்டை காணலாம்.

எடுத்துக்காட்டு:

விகிதமுறு எண்கள் முழு எண்களின் விகிதமாகும். அவை மெய் எண்களின் முக்கிய துணை வர்க்கமாக அமைகிறது. 8/3 அல்லது 19/23 போன்ற விகித முறு எண்களை பின்வருமாறு எழுதலாம்.

< தொகுதி >/< பகுதி >

தொகுதி மற்றும் பகுதி ஆகிய இரண்டும் முழு எண்ணின் இடம் மதிப்புகளாகும். இந்த இரண்டு பகுதிகளும் விகிதமுறு எண்களை சரியாக கனிப்பதற்கு பயன்படுகிறது. முழு எண்களை வகுக்கும் போது முழு எண்ணின் துல்லியம் இழந்து தோராயமான மிதவை எண் கிடைக்கிறது.

8/3 =2.6666666666666665

எனினும், சரியான முறையில் விகித முறு எண்களை குறிப்பதற்கு தொகுதி, பகுதி ஆகிய இரண்டையும் இணைந்து கொடுக்க வேண்டும்.

செயற்கூறு கருத்துக்களைக் கொண்டு, நிரலின் சில பகுதிகளை செயல்படுத்துவதற்கு முன்னதாக நாம் ஒரு நிரலின் செயல்திறனை மேம்படுத்திக் கொள்ள முடியும். நாம், ஒரு விகிதமுறு என்ணை, தொகுதி மற்றும் பகுதி கொண்டு அமைக்க முடியும் என்ற கருதிக்கொள்வோம். அதே போன்று, கொடுக்கப்பட்ட விகிதமுறு எண்ணிற்கு நாம் தொகுதி மற்றும் பகுதியை தேர்ந்தெடுக்க முடியும். மேலும் கண்ஸ்டரக்டர் மற்றும் செலக்டர்ஸ் கொண்டும் நிறைவேற்றலாம்.

இதில் விஸ்புல் திங்கிங் என்ற சக்தி வாய்ந்த செயல் யுக்தியை பயன்படுத்தி நிரல் வடிவமைக்கப்படுகின்றது. நாம் இதுவரை, ஒரு விகிதமுறு எண் எவ்வாறு அமைக்க வேண்டும் என்றோ அல்லது ஒரு கண்ஸ்டரக்டர் மற்றும் செலக்டர் எவ்வாறு செயல்படுத்தப்படுகின்றது என்று நாம் பார்க்கவில்லை.

குறிப்பு

விஸ்புல் திங்கிங் என்பது யதார்த்தத் திற்கு பதிலாக நம்பிக்கையின் அடிப்படையில் விருப்பதற்கு ஏற்ப முடிவெடுப்பதாகும்.

எடுத்துக்காட்டு:– விகிதமுறு எண்ணின் ADT

- - constructor

- - constructs a rational number with numerator n, denominator d

rational(n, d)

- - selector

numer(x) → returns the numerator of rational number x

denom(y) → returns the denominator of rational number y

இப்பொழுது, விகிதமுறு எண்களின் செயல்பாடுகள் செலக்டார் செயற்கூறுகள் நியூமர் மற்றும் மீனாம் கொண்டும் மற்றும் கண்ஸ்டரக்டர் செயற்கூறு ரேஸனல் கொண்டும் வரையறுக்கப்படுகிறது. ஆனால் இச்செயற்கூறுகள் வரையறுக்கப்படவில்லை. நமது தேவை என்னவென்றால், தொகுதி மற்றும் பகுதியை இணைத்து கூட்டு மதிப்பை உருவாக்கும் வழிமுறையாகும்.

மேலே காணும் கண்ஸ்டரக்டர்ஸ் மற்றும் செலக்டர்ஸ் கொண்டு விகிதமுறு எண்களை உருவமைக்கும் போலிக் குறிமுறை

x,y:=8,3

rational(*x*,y)

numer(x)/denum(y)

- - output : 2.6666666666666665

2.5 Lists,Tuples

தரவு அருவமாக்கினை ஸ்திரமுடன் செயல்படுத்த, பைதான் போன்ற மொழிகள் பேர்ஸ் எனும் கூட்டு அமைப்பை வழங்குகிறது. இவை லிஸ்ட் மற்றும் டூபுல்ஸ் ஆகியவற்றால் உருவாக்கப்படுகிறது. பேர்ஸ்சை, லிஸ்ட் கொண்டு எளிதாக செயல்படுத்தலாம்.

2.5.1 List

List அமைப்பு கோவைகளை சதுர அடைப்புக்குறிக்குள் காற்புள்ளியில் பிரிக்கப்பட்டிருக்கும். இவ்கோவைகளை ***List literal*** என்று அழைக்கப்படுகிறது. ***List*** பல மதிப்புகளை சேமிக்கும் இம்மதிப்புகள் எவ்வகையாகவும் இருக்கலாம் அல்லது மற்றொரு லிஸ்டாகவும் இருக்கலாம்.

எடுத்துக்காட்டு List is [10, 20].

list-ன் உறுப்புகளை இரு வழியில் அணுகலாம். முதல் வழி, அதிகம் பயன்படுத்தப்படும் பன்மடங்கு மதிப்பிருத்தல். இம்முறையில், list-ன் உறுப்புகள் பிரிக்கப்பட்டு, அனைத்து உறுப்புகளும் வேறு பெயர்களுடன் இணைக்கப்படுகிறது.

lst := [10, 20]

x, y := lst

மேலே காணும் எடுத்துக்காட்டில், xயின் மதிப்பு 10 என்றும், Y யின் மதிப்பு 20 என மதிப்பிருத்தப்படும்.

இரண்டாம் முறையில் லிஸ்டின் உறுப்புகள், உறுப்புகள் தேர்வு மூலம் அணுகப்படுகிறது. செயற்குறிகள் சதுர அடைப்புக்குறிக்குள் கொடுக்கப்பட்டிருக்கும். லிஸ்ட் லிட்டரல்ஸ் போல் அல்லாது கோவையில் ஒரு சதுர அடைப்புக்குறியை தொடர்ந்து வரும் மற்றொரு சதுர அடைப்புக்குறி லிஸ்டின் மதிப்புகளாக எடுத்துக்கொள்ளப்படுவதில்லை. மாறாக முந்தய கோவையின் உறுப்பை மதிப்பாக தேர்வு செய்யும்.

lst[0]

10

lst[1]

20

மேலே காணும் எடுத்துக்காட்டினை கணித முறையில் set போன்று அமைக்கலாம்.

lst[(0, 10), (1, 20)] - where

(0, 10) → Index position, value

(1, 20) → Index position, value

இரு மதிப்புகளை ஒன்றாக இணைக்கும் முறையை *Pairs* என்று அழைக்கிறோம். *List* அதிகமாக பயன்படுத்தப்படும் முறையாகும். எனவே, ***List***, ***Pairs*** என்று அழைக்கப்படுகிறது.

2.5.1.1 List கொண்டு விகிதமுறு எண்களை உருவமைப்பு.

போலிக் குறிமுறையில், தொகுதி, பகுதி ஆகிய இரு முழு எண்களின் Pairs கொண்டு விகிதமுறு எண்களை உருவமைக்கலாம்.

```
rational(n, d):
    return [n, d]
numer(x):
    return x[0]
denom(x):
    return x[1]
```

2.5.2 Tuple

Pairs என்பது இரு தரவு பகுதிகளை சேமிக்கும் கலவை தரவு என்பதை நினைவில் கொள்ள வேண்டும். இதுவரை இருவேறு முறைகளில் pairs தரவு வகை உருவமைப்பை பார்த்தோம். முதல் வழி, List கொண்டு உருவமைத்தல் மற்றும் இரண்டாம் வழி pairs கொண்டு வடிவமைத்தல் ஆகும்.

tuple என்பது அடைப்புக்குறிக்குள் கொடுக்கப்பட்டுள்ள தொடர் மதிப்புகளை காற்புள்ளியில் பிரிக்கப்பட்டிருக்கும். tuple என்பது List போன்றதாகும். இவை இரண்டிற்கும் உள்ள வேறுபாடு என்னவென்றால் tuple-ல் கொடுக்கப்பட்ட உறுப்புகளை மாற்ற முடியாது, ஆனால் List-ல் இடம்பெற்றுள்ள உறுப்புகளை மாற்றலாம்.

எடுத்துக்காட்டு colour= ('red', 'blue', 'Green')

2.5.2.1 Tuple-ஐ Pair உருவமைத்தல்

```
nums := (1, 2)
nums[0]
1
nums[1]
2
```

சதுர அடைப்புக்குறி குறியீடு, பேர்ஸில் சேமிக்கப்பட்டுள்ள தரவுகளை அணுகுவதற்கு பயன்படுத்துகிறது என்பதை கவனிக்கவும். தரவுகள் பூஜ்யம் சுட்டு வரிசைப்பட்டுள்ளதால், முதல் உறுப்பு nums[0] என்றும் இரண்டாம் உறுப்பு nums[1] என்று அணுகப்படும்.

2.6 ஸ்டரக்சர்லில் தரவு அருவமாக்கம்

List தரவு அருவமாக்கம் மூலம் நினைவக செல்கள் தொகுப்பிற்கு பெயரிட உதவுகின்றது என்பதை நாம் முன்னதாக அறிந்துள்ளோம். உதாரணமாக மாஸ்டர் மைண்ட் விளையாட்டில், ஆட்டக்காரர்கள் யூகிக்கும் நான்கு நிரங்களின் பட்டியலை கண்காணிக்க வேண்டும். நான்கு வெவ்வேறு மாறிகளுக்கு (color1, color2, color3, மற்றும் color4) மாற்றாக 'Predict', என்ற ஒற்றை மாறியை பயன்படுத்தலாம்.

எடுத்துக்காட்டு, Predict =['red', 'blue', 'green', 'green']

List பல விருப்பப்படி பொருளின் பல்வேறு பகுதிகளை பெயரிட அனுமதிப்பதில்லை. Predict முறையில் நீங்கள் பகுதிகளை பெயரிட தேவையில்லை. அட்டவணையைக் கொண்டு ஒவ்வொரு நிரத்தினையும் பெறலாம்.

ஆனால் மிகவும் சிக்கலான விஷயத்தில் ஒரு நபர் எடுத்துக்கொண்டால், அதில் பல உருப்படி பொருளில், அனைத்து உருப்புகளும் பெயரிடப்பட்டிருக்கும். முதல் பெயர், கடைசி பெயர், அடையாளம், மின்னஞ்சல் ஒரு நபரை லிஸ்ட் கொண்டு குறிப்பிடலாம்:

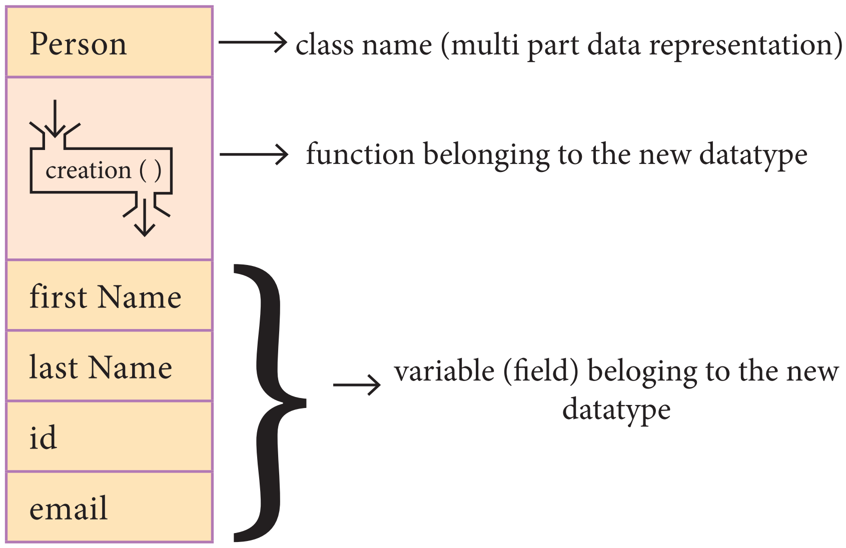
person=['Padmashri', 'Baskar', '994-222-1234', 'compsci@gmail.com']

ஆனால் இது போன்ற குறியீடு ஒவ்வொரு உருப்படியும் எதை குறிக்கின்றது என்று வெளிப்படையாக குறிப்பதில்லை.

இச்சிக்கலை list பயன்படுத்துவதற்கு மாற்றாக structure அமைப்பை (OOP மொழியில் இது இனக்குழு அமைப்பு என்று அழைக்கப்படுகிறது) கொண்டு பல உருப்படி பொருளில் அனைத்து உருப்புகளும் பெயரிடப்படும் (பெயர் கொடுக்கப்பட்டு பின்வரும் போலிக்குறிமுறையை காணவும்:

```
class Person:
        creation( )
        firstName := " "
        lastName := " "
        id := " "
        email := " "
```

Person எனும் புதிய தரவு வகையை படவடிவில் உருவமைக்கலாம்.



பின்வருபவனவற்றை main() கொண்டிருக்கும்	
p1:=Person()	பொருள் உருவாக்கும் கூற்று.
firstName := " Padmashri "	-- firstName எனும் புலத்திற்கு Padmashri என மதிப்பிருத்தப்படுகிறது.
lastName :="Baskar"	lastName எனும் புலத்திற்கு Baskar என மதிப்பிருத்தப்படுகிறது.
id :="994-222-1234"	id எனும் புலத்திற்கு 994-222-1234 என மதிப்பிருத்தப்படுகிறது.
email="compsci@gmail.com"	email எனும் புலத்திற்கு compsci@gmail.com என மதிப்பிருத்தப்படுகிறது.
- - output of firstName : Padmashri	

Chapter 2.indd 18 30-01-2020 19:08:01

இனக்குழு கட்டமைப்பு ஒரு நபரை குறிக்கும் பல பகுதி பொருளுக்கு வடிவம் வரையறுக்கிறது. இந்த வரையறுப்பு புதிய தரவு வகையை சேர்க்கிறது. இதில் நபர் என்பது வகையாகும். ஒரு முறை வரையறுத்தலின், நாம் புதிய மாறிகளை உருவாக்கலாம். இந்த எடுத்துக்காட்டில் Person என்பது இனக்குழு அல்லது வகையாகும. p1 என்பது தனிப்பட்ட குக்கீ குறிக்கின்றது. குக்கீ கட்டர் கொண்டு பல குக்கீகளை தயாரிக்கலாம். அதுபோன்று இவ்இனக்குழுவைக் கொண்டு பல பொருள்களை உருவாக்கலாம்.

இதுவரை, இனக்குழு தொடர்புடைய தரவுகளைக் கொண்டு தரவு அருவமாக்கத்தை எவ்வாறு வரையறுக்கிறது என்று பார்த்தோம். ஓர் இனக்குழு என்பது தரவுகளை மட்டுமில்லாது செயற்கூற்றினையும் தன்னுடன் கொண்டிருக்கும். இவ்வகை செயற்கூறுகள், இனக்குழுவிற்கு கீழ்படந்தவை, ஏனெனில் இவை இனக்குழுவின் தரவுகளை வைத்து செயல்படுகிறது. எடுத்துக்காட்டு Person பொருளின் தரவுகளை மாற்றம் செய்தல் அல்லது ஆய்வு செய்தல் எனவே, இனக்குழு என்பது தரவு மற்றும் அத்தரவின் மீது செயலாற்றும் செயற்கூறுகளை கொண்ட தொகுப்பாகும். முன்னதாக பார்த்த எடுத்துக்காட்டு மற்றும் விளக்கத்தின் மூலம் அருவமாக்கம் என்பது சிக்கலான தரவுகளை எளிய முறையில் கையாள உதவுகின்றது என்பதை உணரலாம்.

Therefore we can define ***a class as bundled data and the functions that work on that data***. From All the above example and explanation one can conclude the beauty of data abstraction is that we can treat complex data in a very simple way.

நினைவில் கொள்க

- ஒரு பொருளின் இயக்கம் அருவமாக்கம் தரவு வகையின் (ADT) அல்லது இனக்குழு தொடர் மதிப்பு மற்றும் தொடர் செயல்பாடுகள் மூலம் வரையறுக்கப்படுகிறது.
- ADT வரையறை, எத்தகைய செயல்பாடுகள் செயப்பட வேண்டும் என்பதை குறிக்கும், ஆனால் இச்செயல்பாடுகள் எவ்வாறு செயல்படுத்தப்படும் என்று குறிப்பிடப்படுவதில்லை.
- ADT தரவுகள் நினைவகத்தில் எவ்வாறு சேமிக்கப்படுகிறது என்றோ செயல்பாடுகளை நிறைவேற்றும் வழிமுறைகளை குறிப்பிடுவதில்லை.
- கண்ஸ்டரக்டர் செயற்கூறுகள் அருவமாக்கம் தரவு வகையை உருவாக்குகின்றது.
- செலக்டர்ஸ் செயற்கூறுகள் தரவு வகையிலிருந்து தகவல்களை மீட்டெடுக்க பயன்படுகிறது.
- கான்கீரிட் தரவு வகை (CDT's) அல்லது ஸ்டக்சர்ஸ் எளிய கருத்தினை செயல்படுத்த உதவுகிறது.
- அருவமாக்கம் தரவு வகை (ADT's) ஒரு கருத்து எவ்வாறு செயல்படுத்தப்படுகின்றது என்பதை தவிர்த்து, அக்கருத்து பற்றிய உயர் நிலை பார்வை (மற்றும் பயன்பாடு) வழங்குகிறது.
- காண்கீரிட் தரவு வகையின் உருவமைப்பு அறியப்பட்டது, ஆனால் அருவமாக்கம் தரவு வகையின் உருவமைப்பு அறியப்படாது.
- Pairs எனும் கலவை அமைப்பு, List அல்லது Tuples கொண்டு உருவாக்கப்படுகிறது.
- List கோவை, சதுர அடைப்புக்குறிக்குள் காற்புள்ளியால் பிரிக்கப்பட்டு உருவமைக்கப்படும்.
- List-ல் உள்ள உருப்புகளை இருவழியில் அணுகலாம். முதல் வழிமுறை, பல்வேறு மதிப்பிருத்தல் முறை மற்றும் இரண்டாம் வழிமுறை உறுப்பு தேர்ந்தெடுப்பு செயற்கூறி ஆகும்.
- இரு மதிப்புகளை ஒன்றாக பிணைப்பது பேர்ஸ் என்று கருதப்படுகிறது.
- பல் உருப்பு பொருளின் பல்வேறு பகுதிகளை பெயரிட List அனுமதிப்பதில்லை.

GXXBD2

மதிப்பீடு

பகுதி–அ

சரியான விடையை தேர்ந்தெடுத்து எழுதுக (1 மதிப்பெண்)

1. பின்வரும் எந்த செயற்கூறு அருவமாக்கம் தரவு வகையை உருவமைக்கப் பயன்படுகிறது ?

   (அ) Constructors (ஆ) Destructors (இ) recursive (ஈ)Nested

2. பின்வரும் எந்த செயற்கூறு தரவு வகையில் இருந்து தகவல்களை மீட்டெடுக்கும்?

   (அ) Constructors (ஆ) Selectors (இ) recursive (ஈ)Nested

3. வரிசைப்படுத்தப்பட்ட உருப்புகளை மாற்றக்கூடிய தரவு கட்டமைப்பு

   (அ) Built in (ஆ) List (இ) Tuple (ஈ) Derived data

4. மாற்ற செய்ய முடியாத பொருளின் தொடர் வரிசை

   (அ) Built in (ஆ) List (இ) Tuple (ஈ) Derived data

5. உருவமைப்பு அறியப்பட்ட தரவு வகை எவ்வாறு அழைக்கப்படுகிறது.

   (அ) Built in datatype (ஆ) Derived datatype

   (இ) Concrete datatype (ஈ) Abstract datatype

6. உருவமைப்பு அறியப்படாத தரவு வகை எவ்வாறு அழைக்கப்படுகிறது

   (அ) Built in datatype (ஆ) Derived datatype

   (இ) Concrete datatype (ஈ) Abstract datatype

7. பின்வருவனவற்றில் எது கலவை அமைப்பு?

   (அ) Pair (ஆ) Triplet (இ) single (ஈ) quadrat

8. இரு மதிப்புகளை ஒன்றாக பிணைப்பு எந்த வகை கருதப்படுகிறது.

   (அ) Pair (ஆ) Triplet (இ) single (ஈ) quadrat

9. பின்வருவனவற்றில் எது பல் உருப்பு பொருளின் பல்வேறு பகுதிகளை பெயரிட அனுமதிக்கிறது?

   (அ) Tuples (ஆ) Lists (இ) Classes (ஈ) quadrats

10. பின்வருவனவற்றில் எது கோவைகளை சதுர அடைப்புக்குறிக்குள் வைத்து உருவமைகிறது ?

    (அ) Tuples (ஆ) Lists (இ) Classes (ஈ) quadrats

பகுதி–ஆ

அனைத்து வினாக்களுக்கும் விடையளி (2 மதிப்பெண்)

1. தரவு அருவமாக்கம் வகை என்றால் என்ன?
2. ஆக்கிகள் மற்றும் செலக்டர்கள் வேறுபாடு தருக.

3. Pair என்றால் என்ன? எடுத்துக்காட்டு தருக..
4. List என்றால் என்ன? எடுத்துக்காட்டு தருக
5. Tuple என்றால் என்ன? எடுத்துக்காட்டு தருக.

பகுதி–இ

அனைத்து வினாக்களுக்கும் விடையளி (3 மதிப்பெண்)

1. கான்கிரிட் தரவு வகை மற்றும் அருவமாக்கம் தரவு வகை வேறுபடுத்துக.
2. நிரல் வடிவமைப்பில் பின்பற்றப்படும் யுக்தி எது? யுக்தியை வரையறுக்க.
3. பின்வருவனவற்றில் எது constructors and selectors என்று அடையாளம் காணவும்?
   (a) N1=number() (b) accetnum(n1) (c) displaynum(n1)
   (d) eval(a/b) (e) x,y= makeslope (m), makeslope(n)
   (f) display()
4. list உள்ள உருப்புகளை அணுகும் பல்வேறு வழிமுறைகள் யாவை? எடுத்துக்காட்டு தருக.
5. பின்வருவனவற்றில் எது List, Tuple மற்றும் இனக்குழு (class) என அடையாளம் காண்க.
   (a) arr [1, 2, 34] (b) arr (1, 2, 34) (c) student [rno, name, mark]
   (d) day= ('sun', 'mon', 'tue', 'wed') (e) x= [2, 5, 6.5, [5, 6], 8.2]
   (f) employee [eno, ename, esal, eaddress]

பகுதி–ஈ

அனைத்து வினாக்களுக்கும் விடையளி (5 மதிப்பெண்)

1. தரவு அருவமாக்கம் எவ்வாறு செயல்படுத்துவாய்? எடுத்துக்காட்டுடன் விளக்குக.
2. List என்றால் என்ன? ஏன் List, Pairs என்று அழைக்கப்படுகிறது. எடுத்துக்காட்டுடன் விவரி.
3. பல் உருப்பு பொருளை எவ்வாறு அணுகுவாய் எடுத்துக்காட்டுடன் விளக்கு.

Reference Books

1. *Data structure and algorithmic thinking with python by narasimha karumanchi*
2. *sign and analysis of algorithms by s sridhar*
3. *Data Structures and Algorithms in Python by Goodrich, Tamassia & Goldwasser*
4. *https://www.tutorialspoint.com*