**1.Ի՞նչ է reference-ը։**

Reference –ը այն օբյեկտն է կամ ֆունկցիան, որին նա հղված է։ Այն հայտարարելու համար փոփոխականի առջև դրվում է &(ամպերսանդ) օպերատորը։ Մինչև reference-ի օգտագործելը փոփոխականները պետք է նախապես ինիցիալիզացվեն։ reference-ը միշտ ցույց է տալիս միևնույն հասցեն։ reference-ի և pointer-ի տարբերությունը այն է, որ reference-ը գոյություն ունի ծրագրի կոդում և կոմպիլյացիայի ժամանակ փոխարինվում է իր հասցեով, իսկ pointer-ը փոփոխական է, որում պահվում է հասցեն, որին դիմում են։ **Օր.(reference.c)**

2.**Ի՞նչ է pointer-ը։**

Pointer-ը փոփոխական է, որի արժեքը մեկ այլ փոփոխական է, այսինքն հիշողության ուղիղ հասցեն։ Pointer-ը նախապես պետք է հայտարարվի նախքան նրա մեջ որևէ փոփոխականի հասցե տեղադրելը։ Որպես կանոն, pointer-ի հետ կատավում է 2 հիմնական գործողություն՝ վերագրում և ուղղորդում։ Վերագրման ժամանակ pointer –ին վերագրվում է որոշակի հասցե, իսկ ուղղորդման ժամանակ դիմում ենք այն հասցեին, որը ցույց է տալիս pointer-ը։ pointer-ը կարող է ունենալ նաև 0-ական արժեք կամ NULL, այսինքն տվյալ պահին pointer-ը ոչ մի տեղ ցույց չի տալիս։ Pointer-ը օգտագործվում է համակարգչի հիշողության դինամիկ կառավարման համար։ Այսինքն հիշողության մեջ առանձնացվում են տարածքներ, իսկ այդպիսի փոփոխականները կոչվում են դինամիկ փոփոխականներ։ **Օր(pointer.c)**

3. **Ի՞նչ է Polimorphism-ը։**

# OOP –ի 4 հիմնական սկզբունքներըն են

1. Encapsulation (Ինկապսուլացիա)
2. Inheritance (Ժառանգականություն)
3. Polymorphism (Պոլիմորֆիզմ)
4. Abstraction (Աբստրակցիա)

Պոլիմորֆիզմը նույն օբյեկտը տարբեր միջավայրերում իրեն տարբեր կերպ դրսևորելու հատկությունն է։

1. **Ի՞նչ են, ինչի համար են malloc, calloc-ը ֆունկցիաները։**
2. C- ում «malloc» կամ «“memory allocation» մեթոդը օգտագործվում է նշված չափսի հիշողության 1 մեծ բլոկ առանձնացնեկու համար Այն վերադարձնում է void տիպի pointer, որը կարող է փոխվել ցանկացած տիպի pointer-ի։

***ptr = (int\*) malloc(100 \* sizeof(int));***

Քանի որ int-ը 4 բայթ է, այս գրառումը կհատկացնի 400 բայթ հիշողություն:

ptr-ն վերադարձնում է հիշողության առաջին բայթի հասցեն։

calloc կամ contiguous allocation մեթոդը օգտագորվում է նշված տիպի հիշողության բլոկների առանձնացման համար, յուր. բլոկի համար առանձնացնելով default 0 արժեք։

***ptr = (float\*) calloc(25, sizeof(float));***

Այս գրառումը առանձնացնում էfloat տիպի 25 հարակից բլոկներ ։

**5.Ինչպե՞ս է տեղի ունենում build process-ը. .c ֆայլը դառնում է .exe**

.c ֆայլերի **build process-ը տեղի է ունենում 3 փուլով**

1. **Preprocessing**- Այս պրոցեսը իրականացնում է տեքտային պրեպրոցեսորը։

Մուտքային ֆայլում տեղի են ունենում հետևյալ պրոցեսները

ա. Կոմենտարիաները փոխարինվում են դատարկ տողերով

բ. Կցում են ֆայլերը(#include)

գ. Կատարվում է #if, #ifdef, #elif, #else, #endif, #define դիրեկտիվների մշակում

**file.c🡪file.c՛**

1. **Compilation** –Կոմպիլյացի արդյունքը օբյեկտային կոդն է

**file.c՛🡪file.օ**

1. **Layout** –բոլոր օբյեկտային ֆայլերը հավաքվում են և կազմում ելքային ֆայլը

**file.օ🡪file.exe**

Արդյունքում

**file.c🡪file.c՛🡪 file.օ🡪՚file.exe**

1. **Ինչ՞ է նշանակում static։**

Static տերմինը կարող է օգտագործվել 3 հիմնական կոնտեքստներում

ա) ֆունկցիայի ներսում, այսինքն փոփոխականները ստեղծվում են, երբ ծրագիրը սկսվում է, և ոչնչանում են, երբ ծրագիրը դադարում է։

բ) class-ի ներսում,այսինքն class-ի ստատիկ փոփոխականները պարունակում են ինֆորմացիա, որն անհրաժեշտ է նոր օբյեկտներ ստեղծելու համար։

գ) գլոբալ մակարդակում փոփոխականներ և ֆունկցիաներ հայտարարելիս static տերմինը նշանակում է, որ այն հասանելի չէ այլ ֆայլերից։

**8. ի՞նչ է git-ը։ Ի՞նչ հրամաններ գիտես։**

Git-ը բաց կոդով տվյալների կառավարման ամենատարածված համակարգերից մեկն է, որի վրա հիմնաված են աշխարհի միլիոնավոր պրոյեկտներ։ Այն լրիվ անվճար ծրագրային դաշտ է։

git init – ստեղծվում ենք ռեպազիտորիա

git clone –կլոնավորում ենք ռեպազիտորիա

git status –ստուգում ենք , թե ֆայլերից որոնք չեն գտնվում ռեպազիտորիայում կամ որ ֆայլերն են փոփոխվել

git add file.js– ավելացնում ենք ֆայլ ռեպազիտորիայում

git add . – ավելացնում ենք բոլոր ֆայլերը

git reset file.js – չեղարկում ենք ֆայլի ավելացումը

git reset – չեղարկում ենք բոլոր ֆայլերը

git diff – նայում ենք փոփոխությունները

git ls- ցույց է տալիս բոլոր ֆայլերը տվյալ ֆոլդերում

git commit – վերնագրում ենք փոփոխությունները

git pull – վերբեռնում ենք փոփոխությունները

git push – ավելացնում ենք փոփոխությունները git-ում