SER 334 A Session

SI Session

Thursday, January 18th 2024

7:00 pm - 8:00 pm MST

Agenda

- Pointer Tracing
- Static and Heap Memory
- Structs
- Function Parameters
- Macros

SI Session Expectations

Thanks for coming to the **Enter course** SI session. We have a packed agenda and we are going to try to get through as many of our planned example problems as possible. This session will be recorded and shared with others.

- If after this you want to see additional examples, please visit the drop-in tutoring center.
- We will post the link in the chat now and at the end of the session.
    - tutoring.asu.edu
- Please keep in mind we are recording this session and it will be made available for you to review 24-48 hours after this session concludes.
- Finally, please be respectful to each other during the session.

Interact with us:

Zoom Features

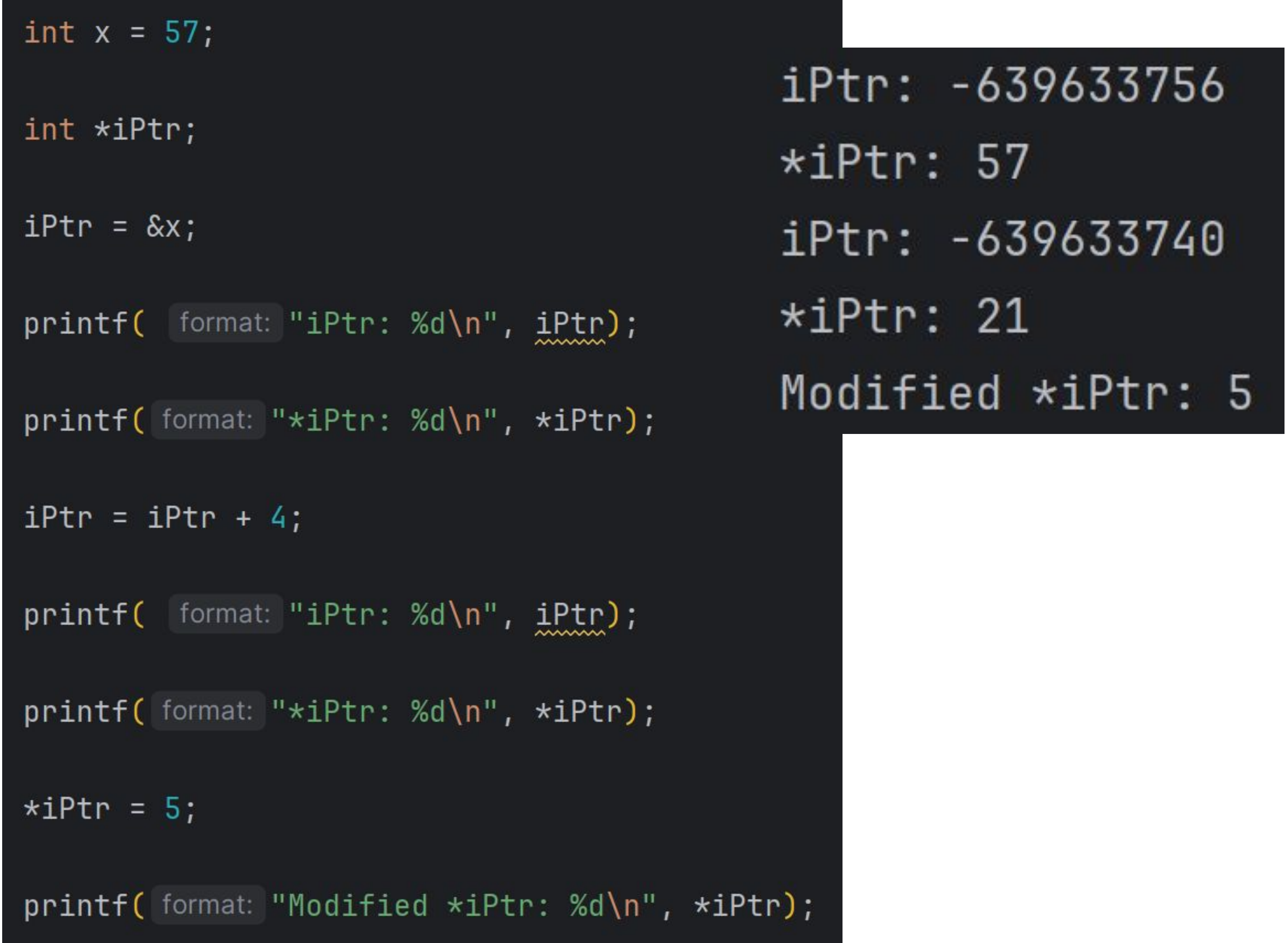
Zoom Chat

- Use the chat feature to interact with the presenter and respond to presenter’s questions.
- Annotations are encouraged

SER 334

Pointer Tracing

Pointer Tracing Practice



```
int x = 57;

int *iPtr;

iPtr = &x;

printf( format: "iPtr: %d\n", iPtr);

printf( format: "*iPtr: %d\n", *iPtr);

iPtr = iPtr + 4;

printf( format: "iPtr: %d\n", iPtr);

printf( format: "*iPtr: %d\n", *iPtr);

*iPtr = 5;

printf( format: "Modified *iPtr: %d\n", *iPtr);
```

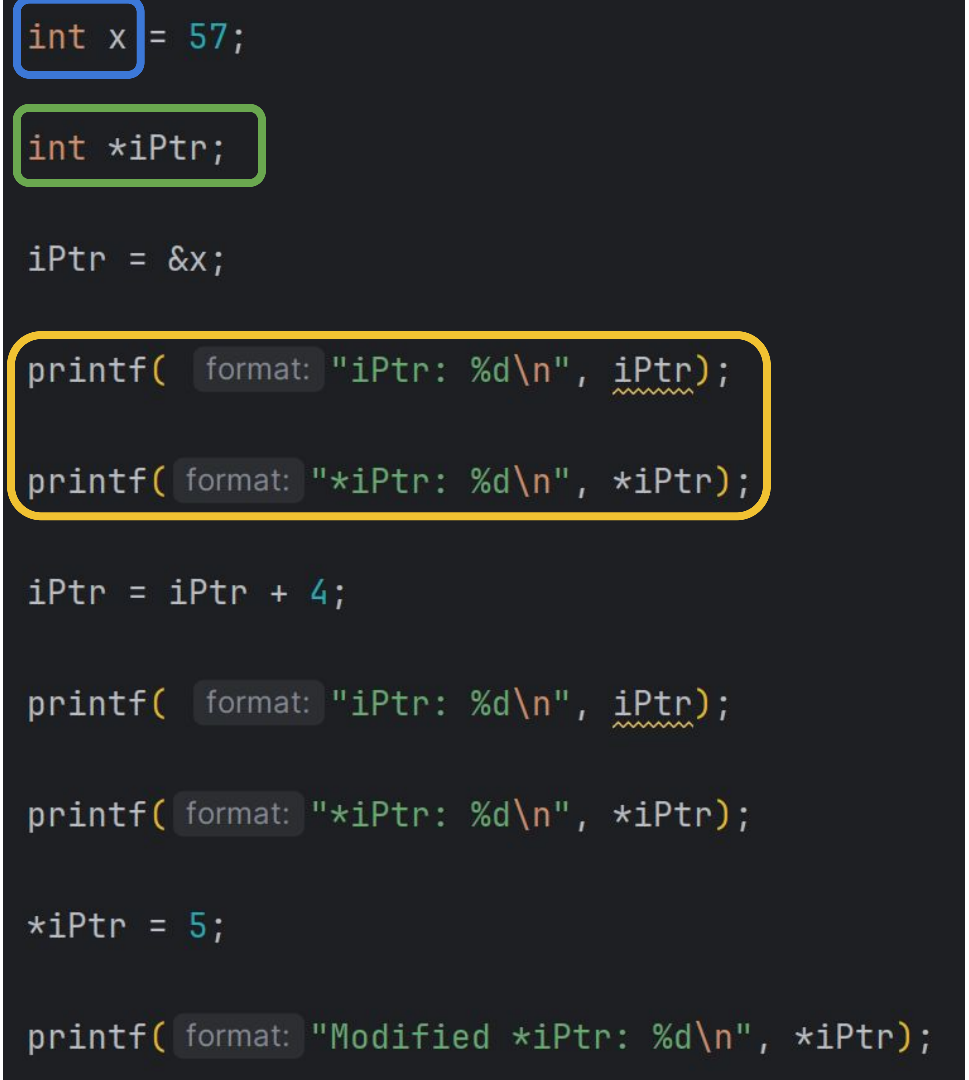
```
iPtr: -639633756
*iPtr: 57
iPtr: -639633740
*iPtr: 21
Modified *iPtr: 5
```

SER 334

Pointer Tracing

Pointer Tracing Practice

Prints:



```
int x = 57;

int *iPtr;

iPtr = &x;

printf( format: "iPtr: %d\n", iPtr);

printf( format: "*iPtr: %d\n", *iPtr);

iPtr = iPtr + 4;

printf( format: "iPtr: %d\n", iPtr);

printf( format: "*iPtr: %d\n", *iPtr);

*iPtr = 5;

printf( format: "Modified *iPtr: %d\n", *iPtr);
```

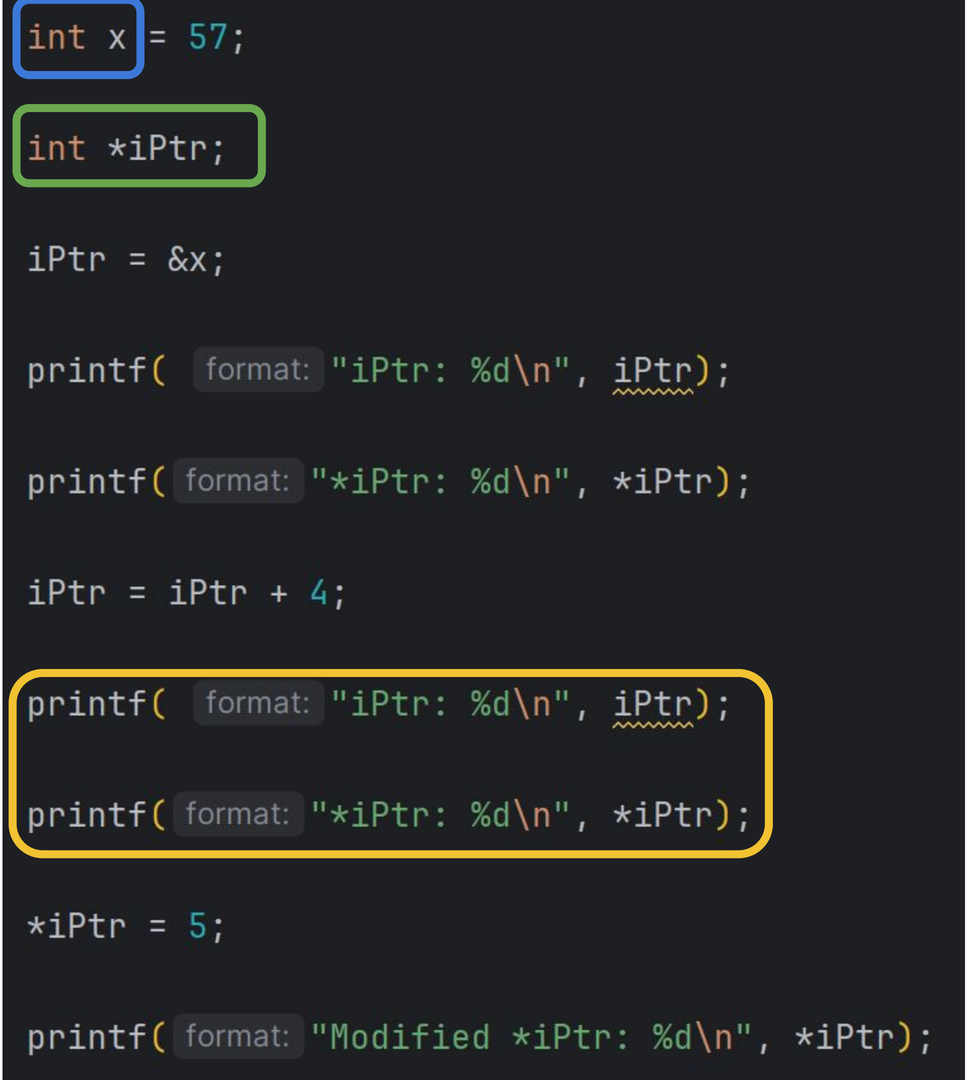
0x1110	57
0x1111	NULL
0x1112	K
0x1113	a
0x1114	t
0x1115	i
0x1116	e
0x1117	\0
0x1118	99
0x1119	0x1118
0x1120	97
0x1121	0x1112
0x1122	j

SER 334

Pointer Tracing

Pointer Tracing Practice

Prints:



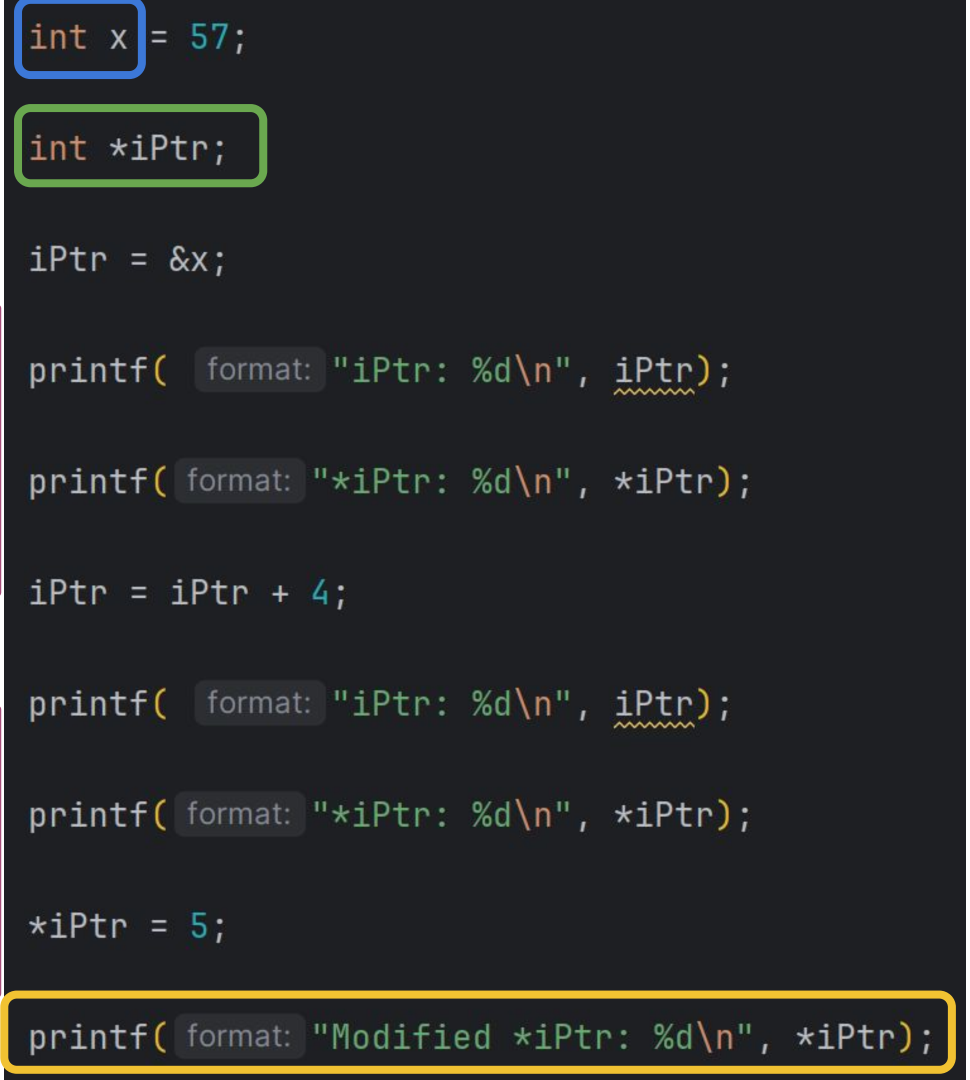
```
int x = 57;

int *iPtr;

iPtr = &x;

printf( format: "iPtr: %d\n", iPtr);

printf( format: "*iPtr: %d\n", *iPtr);

iPtr = iPtr + 4;

printf( format: "iPtr: %d\n", iPtr);

printf( format: "*iPtr: %d\n", *iPtr);

*iPtr = 5;

printf( format: "Modified *iPtr: %d\n", *iPtr);
```

0x1110	57
0x1111	NULL
0x1112	K
0x1113	a
0x1114	t
0x1115	i
0x1116	e
0x1117	\0
0x1118	99
0x1119	0x1118
0x1120	97
0x1121	0x1112
0x1122	j

SER 334
Pointer Tracing

Pointer Tracing Practice

Prints:



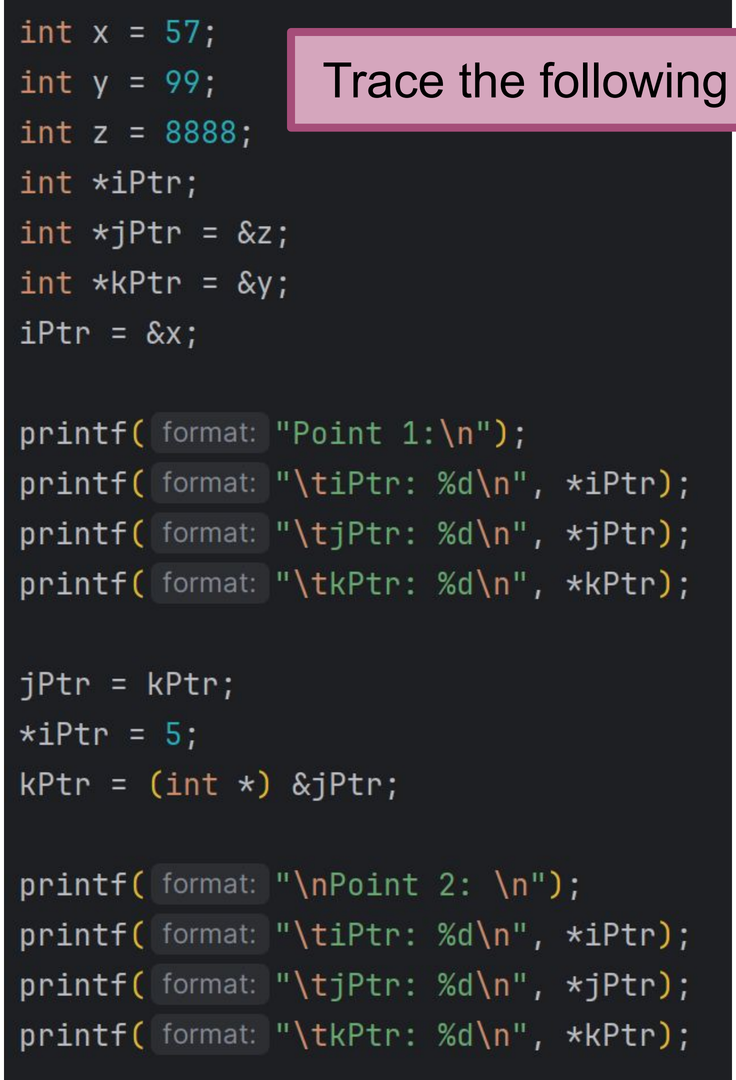
```
int x = 57;

int *iPtr;

iPtr = &x;

printf( format: "iPtr: %d\n", iPtr);

printf( format: "*iPtr: %d\n", *iPtr);

iPtr = iPtr + 4;

printf( format: "iPtr: %d\n", iPtr);

printf( format: "*iPtr: %d\n", *iPtr);

*iPtr = 5;

printf( format: "Modified *iPtr: %d\n", *iPtr);
```

0x1110	57
0x1111	NULL
0x1112	K
0x1113	a
0x1114	5
0x1115	i
0x1116	e
0x1117	\0
0x1118	99
0x1119	0x1118
0x1120	97
0x1121	0x1112
0x1122	j

SER 334

Pointer Tracing

Trace the following code with box and arrow notation

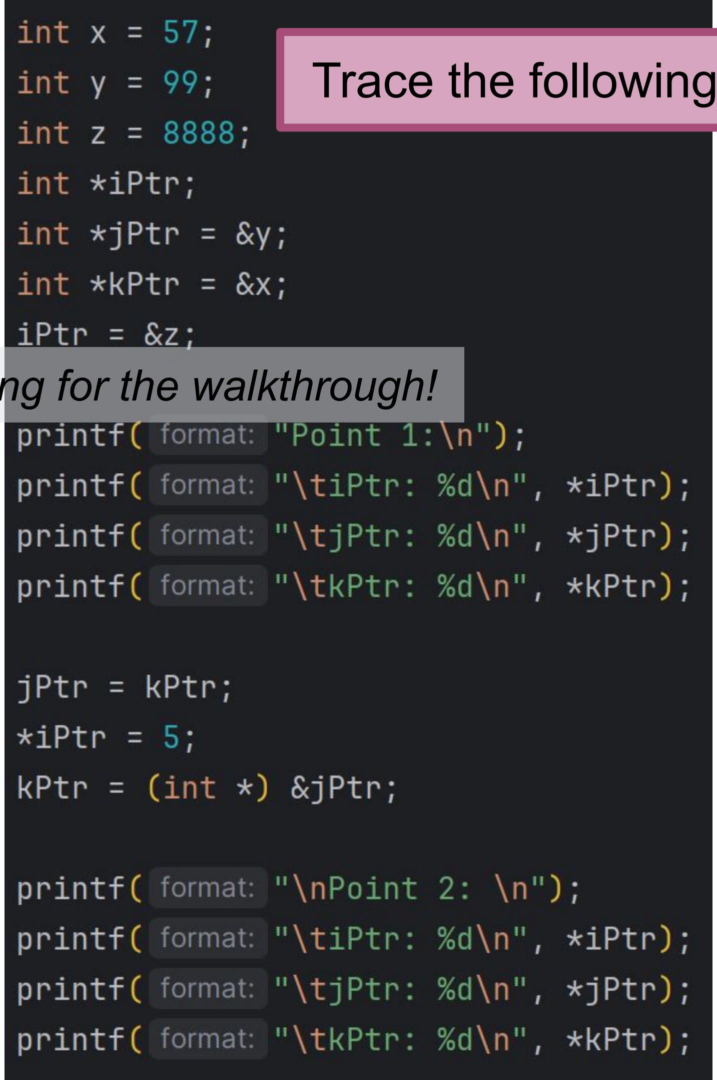


```
int x = 57;
int y = 99;
int z = 8888;
int *iPtr;
int *jPtr = &z;
int *kPtr = &y;
iPtr = &x;

printf( format: "Point 1:\n");
printf( format: "\tiPtr: %d\n", *iPtr);
printf( format: "\tjPtr: %d\n", *jPtr);
printf( format: "\tkPtr: %d\n", *kPtr);

jPtr = kPtr;
*iPtr = 5;
kPtr = (int *) &jPtr;

printf( format: "\nPoint 2: \n");
printf( format: "\tiPtr: %d\n", *iPtr);
printf( format: "\tjPtr: %d\n", *jPtr);
printf( format: "\tkPtr: %d\n", *kPtr);
```

SER 334

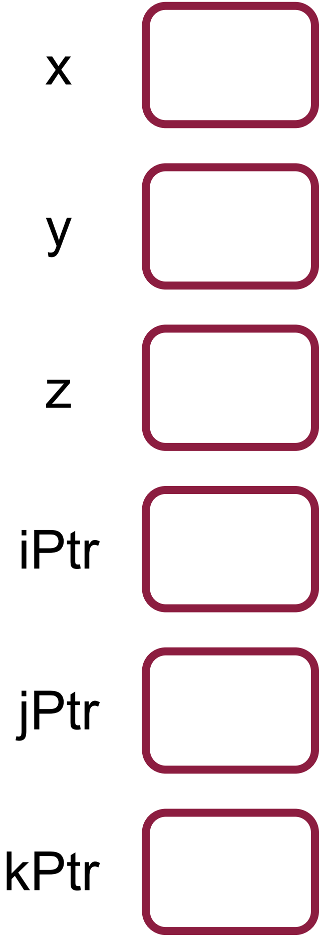
Pointer Tracing

Trace the following code with box and arrow notation



```
int x = 57;
int y = 99;
int z = 8888;
int *iPtr;
int *jPtr = &y;
int *kPtr = &x;
iPtr = &z;

printf( format: "Point 1:\n");
printf( format: "\tiPtr: %d\n", *iPtr);
printf( format: "\tjPtr: %d\n", *jPtr);
printf( format: "\tkPtr: %d\n", *kPtr);

jPtr = kPtr;
*iPtr = 5;
kPtr = (int *) &jPtr;

printf( format: "\nPoint 2: \n");
printf( format: "\tiPtr: %d\n", *iPtr);
printf( format: "\tjPtr: %d\n", *jPtr);
printf( format: "\tkPtr: %d\n", *kPtr);
```

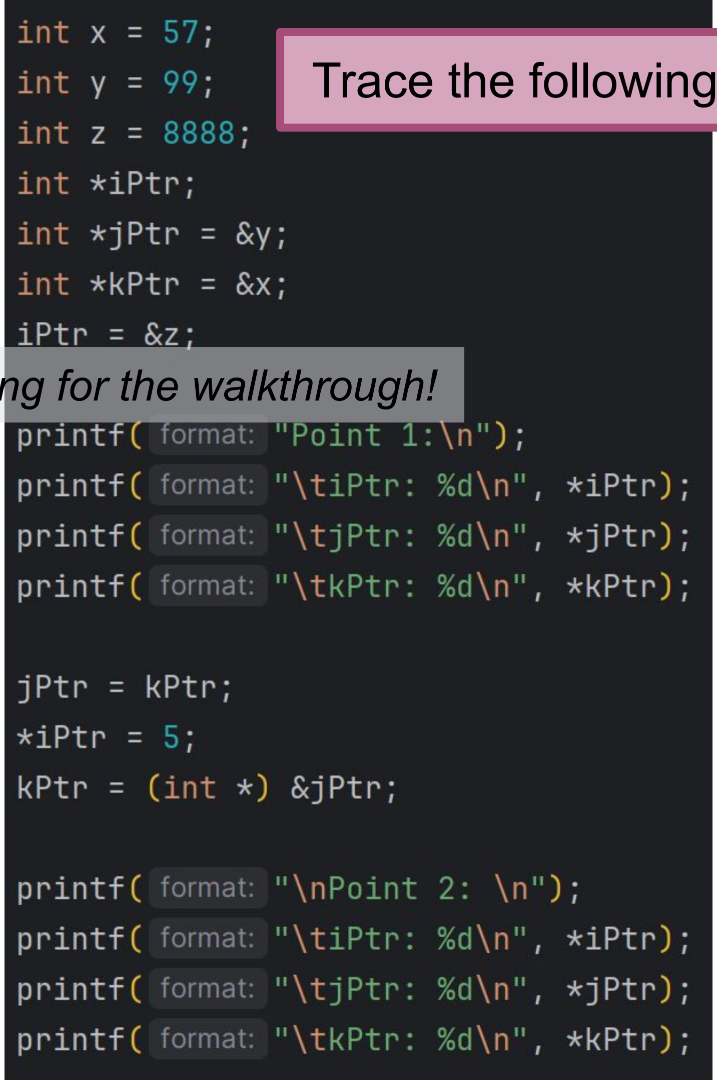
Check out the recording for the walkthrough!



SER 334

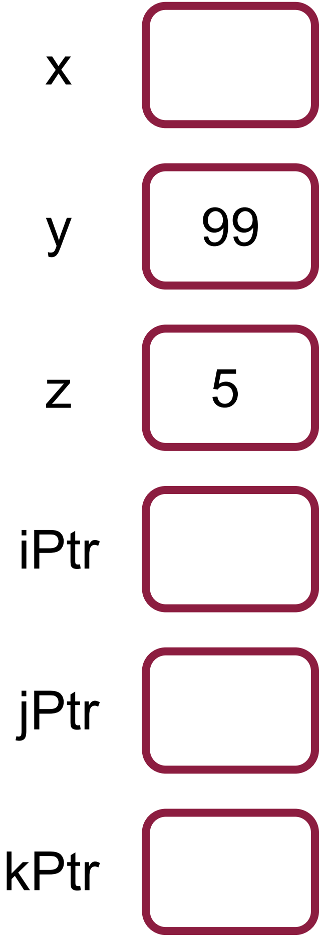
Pointer Tracing

Trace the following code with box and arrow notation



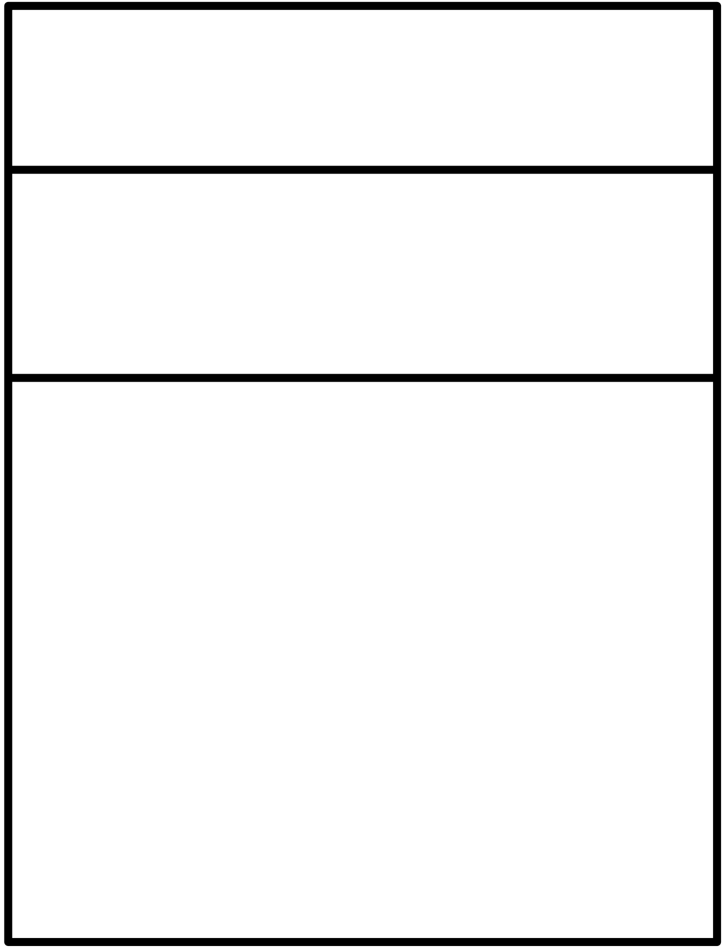
```
int x = 57;
int y = 99;
int z = 8888;
int *iPtr;
int *jPtr = &y;
int *kPtr = &x;
iPtr = &z;

printf( format: "Point 1:\n");
printf( format: "\tiPtr: %d\n", *iPtr);
printf( format: "\tjPtr: %d\n", *jPtr);
printf( format: "\tkPtr: %d\n", *kPtr);

jPtr = kPtr;
*iPtr = 5;
kPtr = (int *) &jPtr;

printf( format: "\nPoint 2: \n");
printf( format: "\tiPtr: %d\n", *iPtr);
printf( format: "\tjPtr: %d\n", *jPtr);
printf( format: "\tkPtr: %d\n", *kPtr);
```

Check out the recording for the walkthrough!



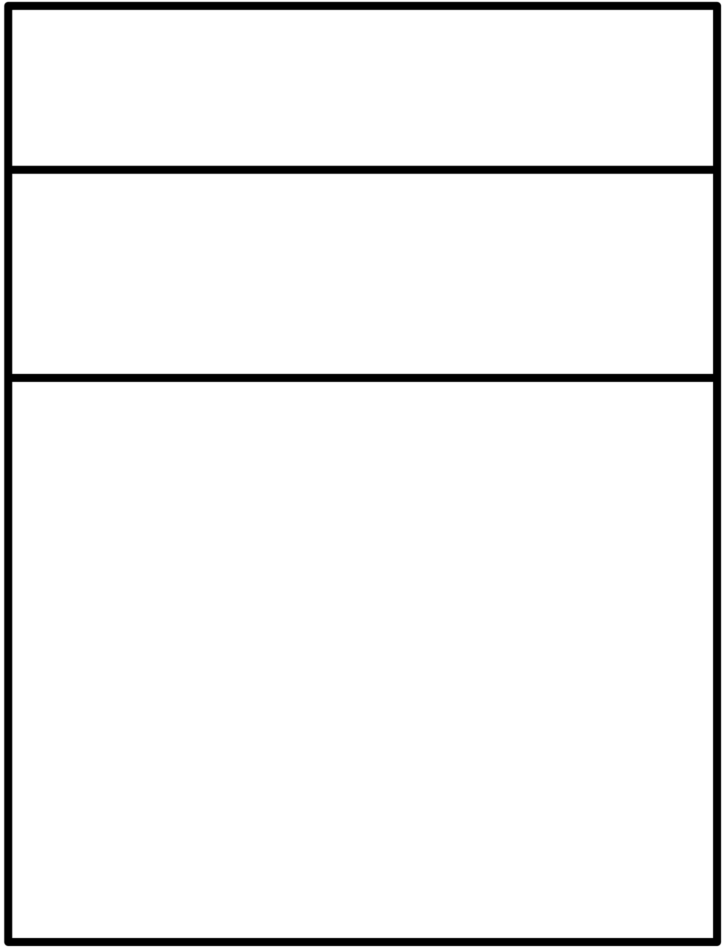
Does anyone want to fill in the memory diagram from here?

Memory

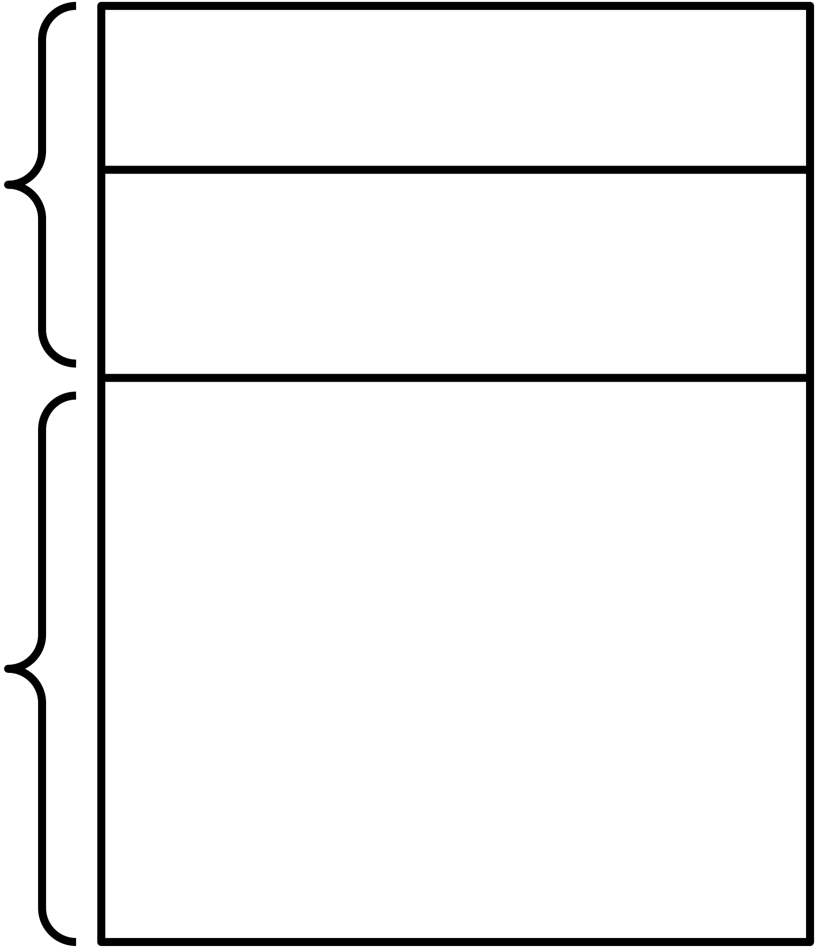


Otherwise I can give some hints…

Memory



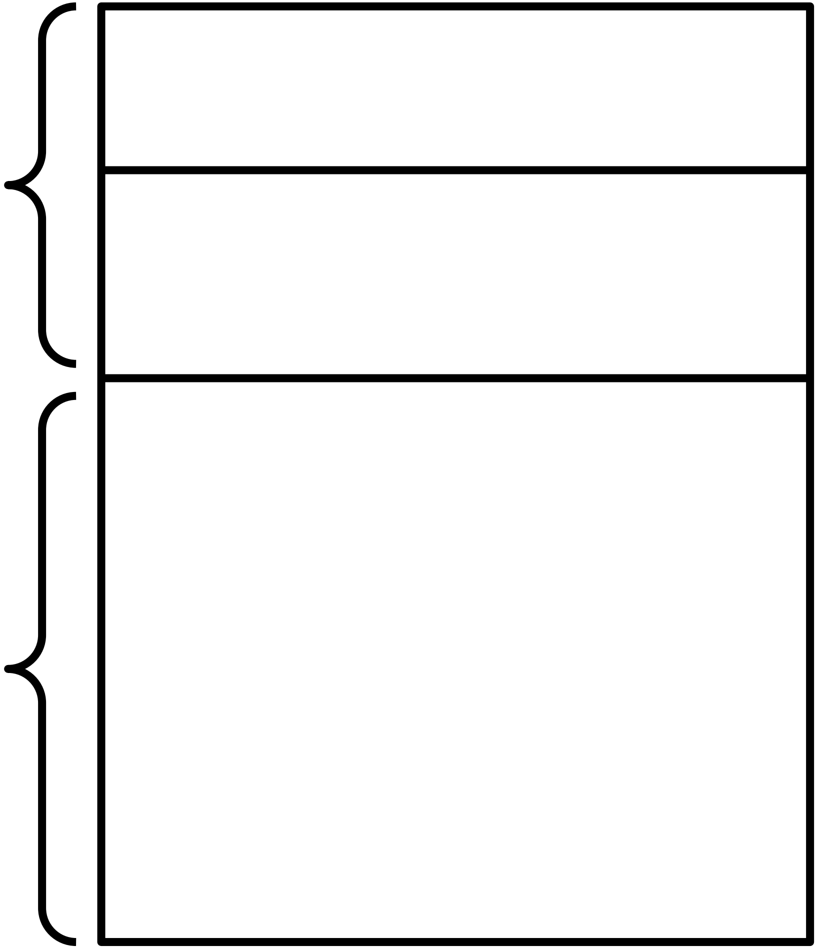
Memory



SER 334

Memory

Memory



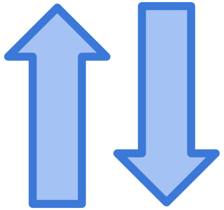
Contents:

Stack

Text/Code

Heap

Global & Static Variables/Objects



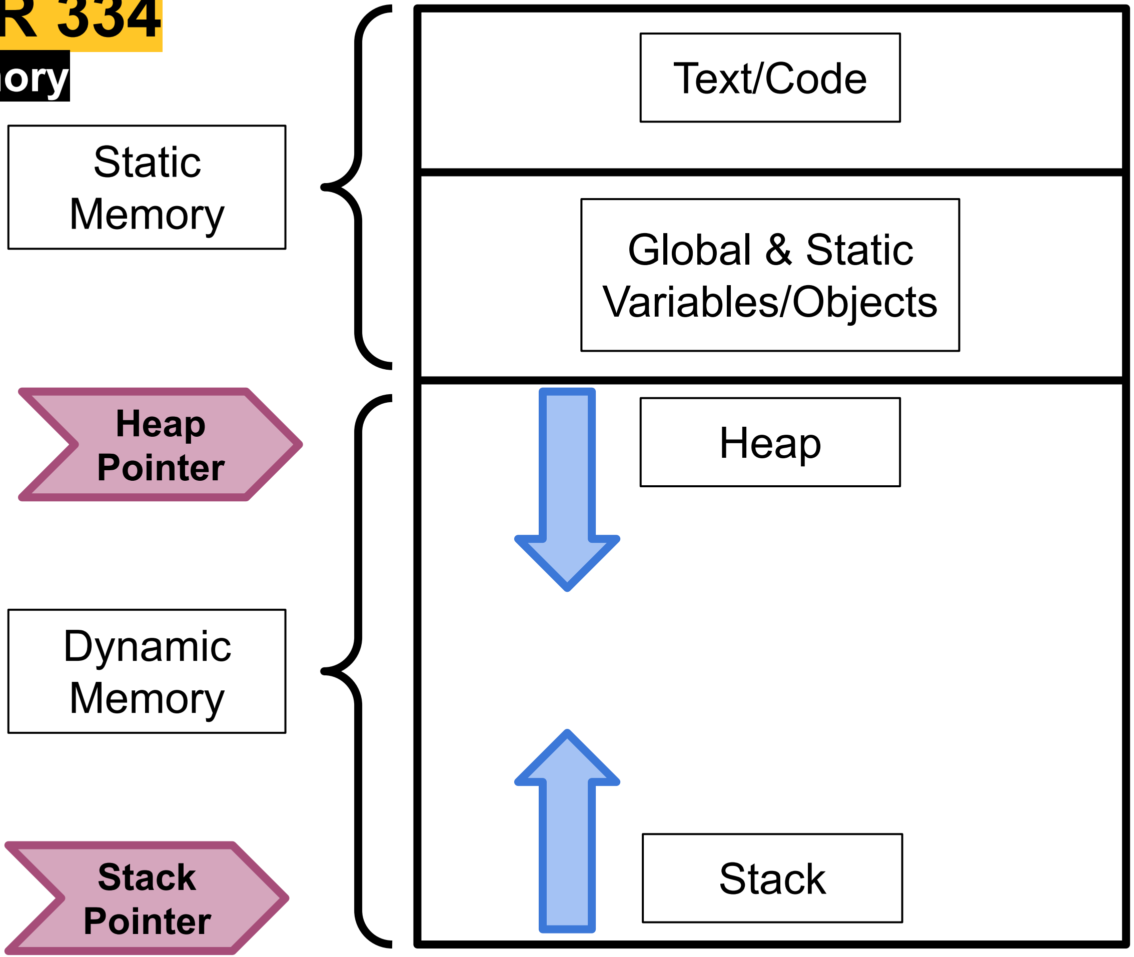
Static Memory

Dynamic Memory

Stack Pointer

Heap Pointer

Memory



Static Memory

Dynamic Memory

Heap Pointer

Stack Pointer

Text/Code

Global & Static Variables/Objects

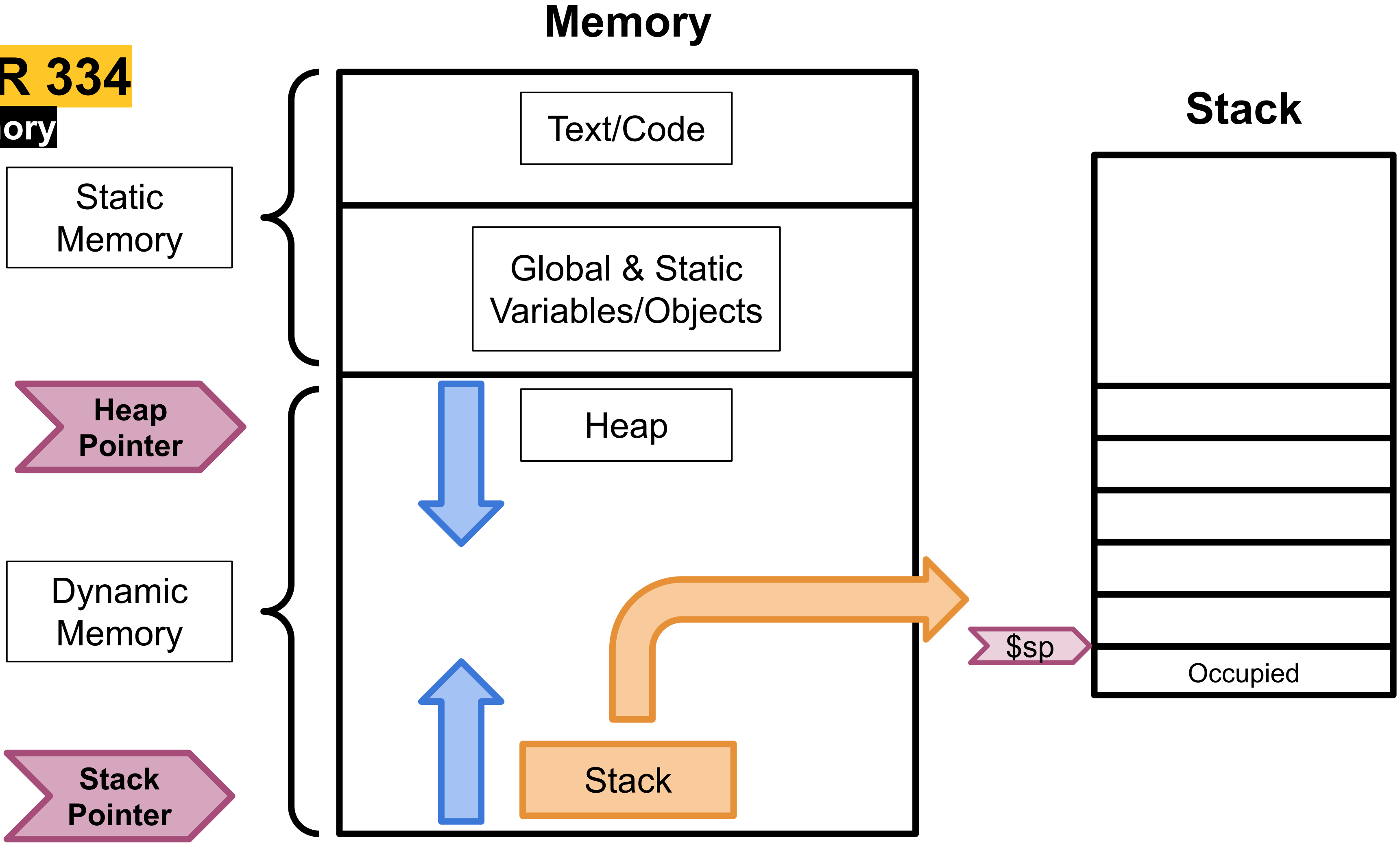
Heap

Stack

Which type of memory is deallocated when it passes out of scope?

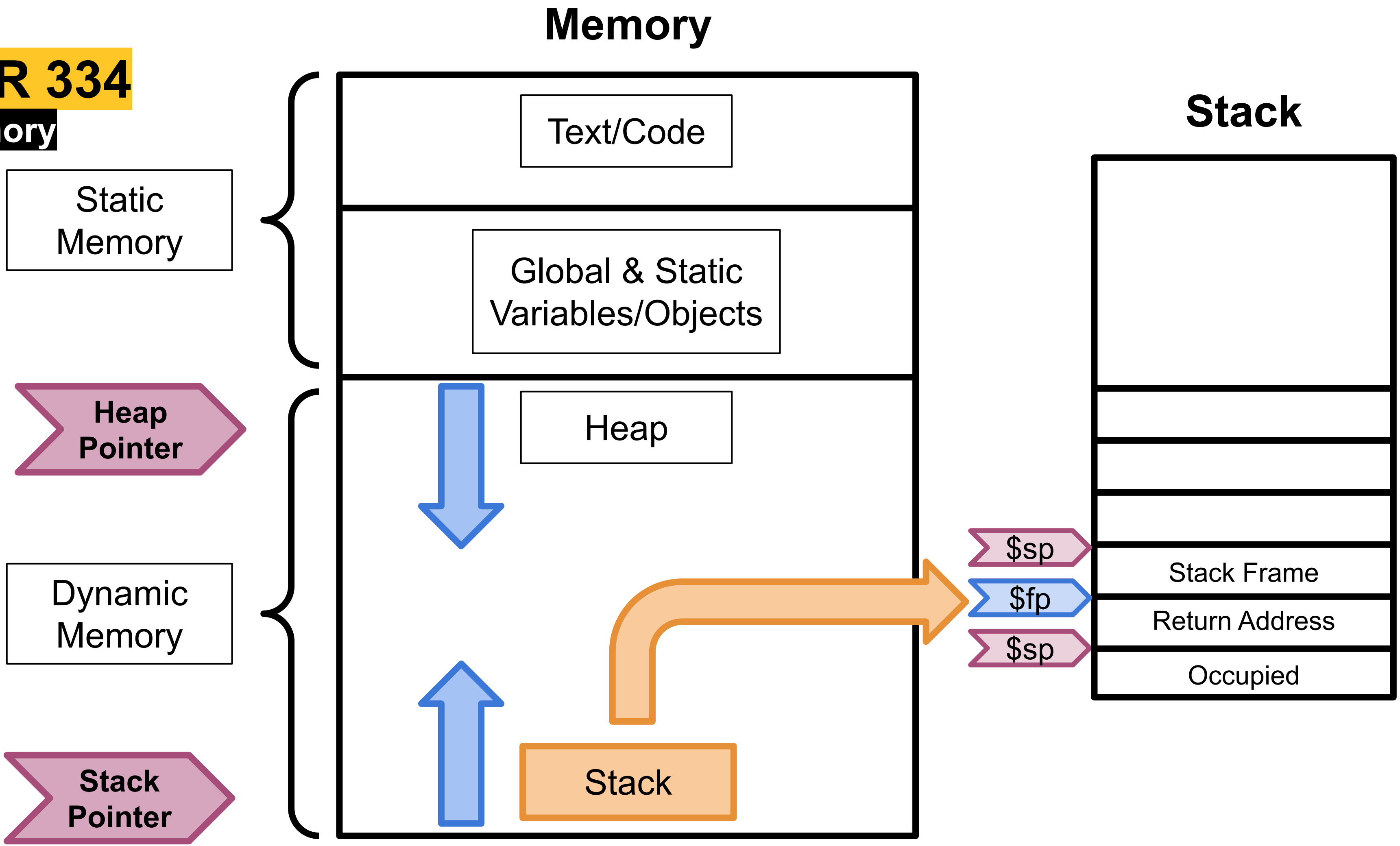
SER 334

Memory



SER 334

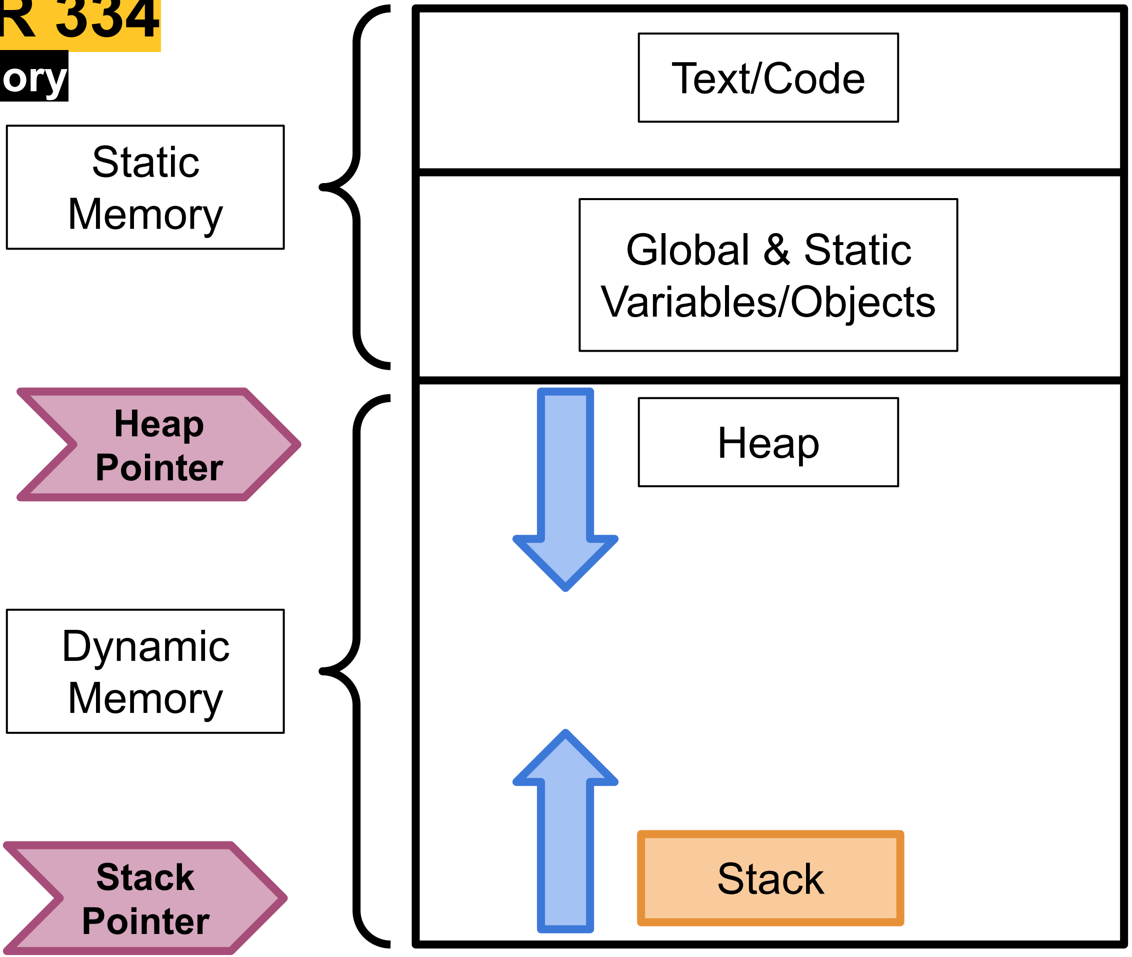
Memory



SER 334

Memory

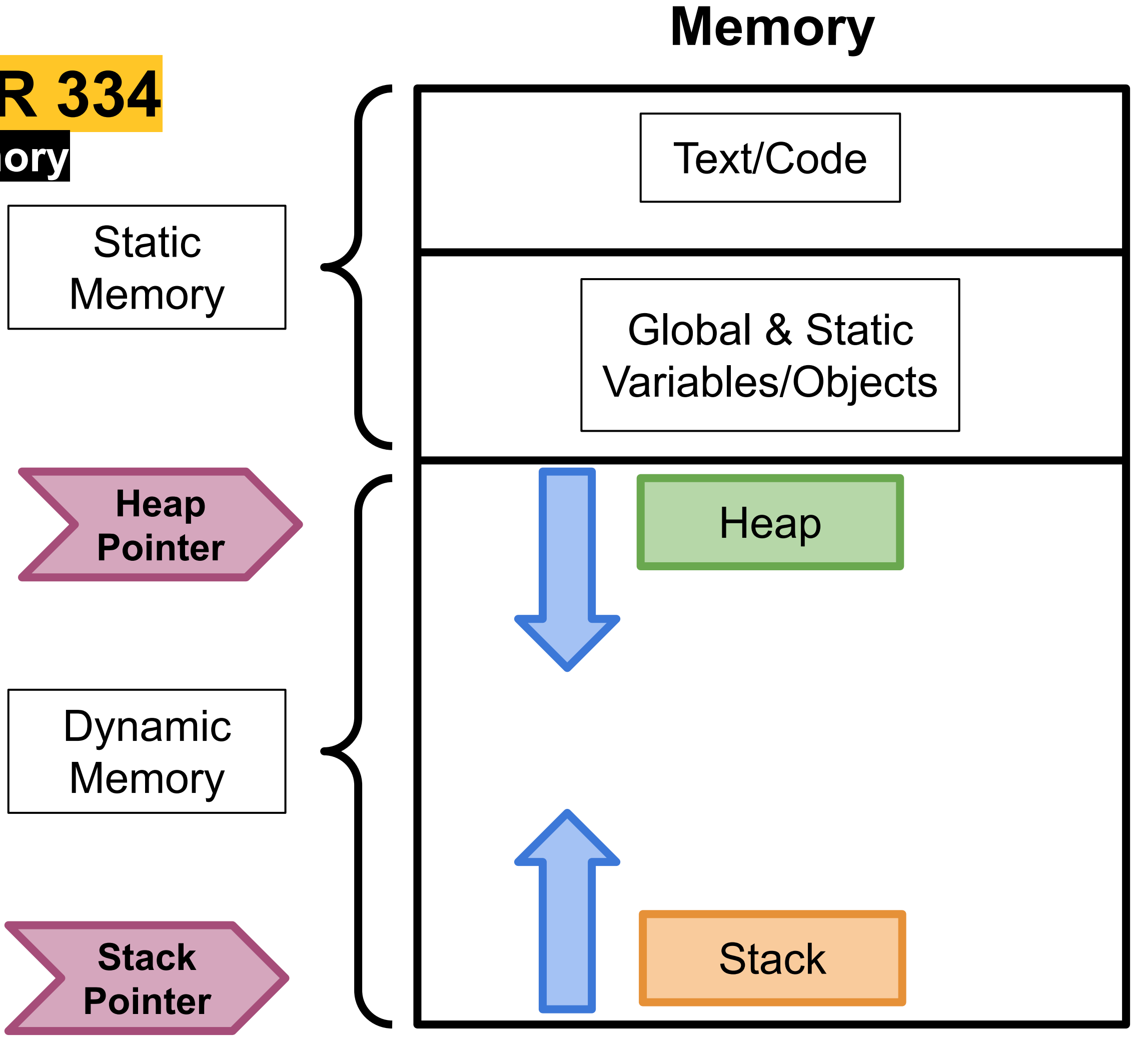
Memory



What about dynamic/run-time memory allocation?

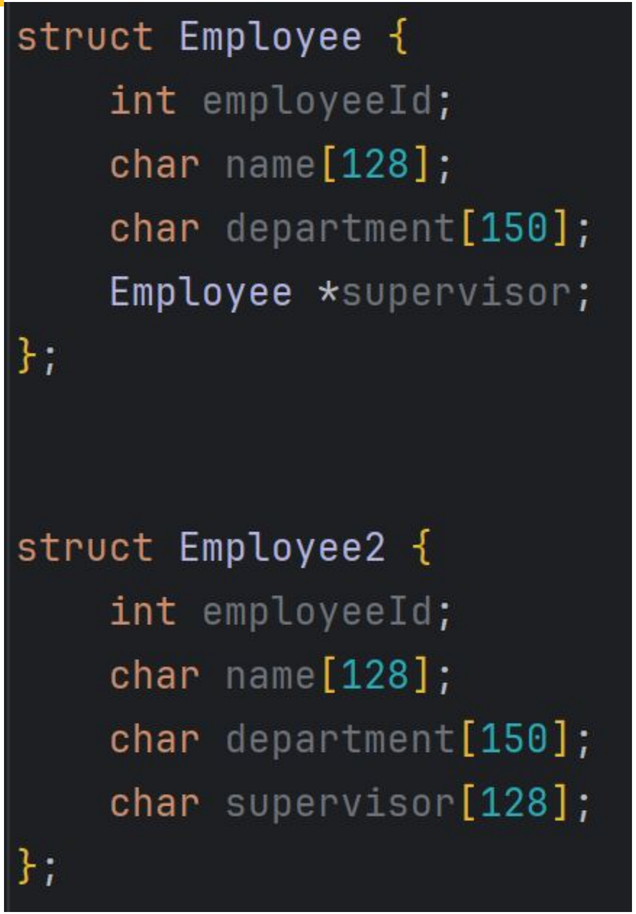
SER 334

Memory



What's different with heap memory?

Which structure uses more memory?

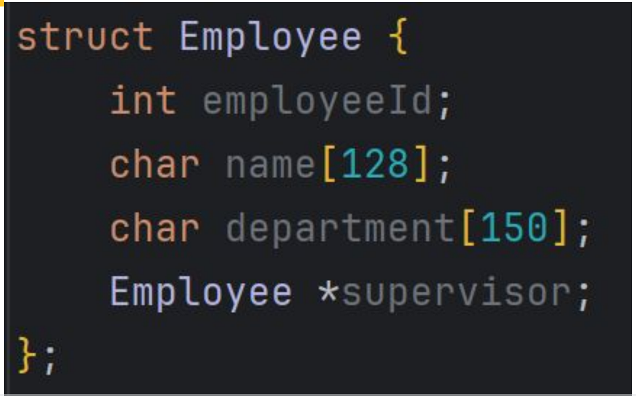


```
struct Employee {
    int employeeId;
    char name[128];
    char department[150];
    Employee *supervisor;
};


struct Employee2 {
    int employeeId;
    char name[128];
    char department[150];
    char supervisor[128];
};
```

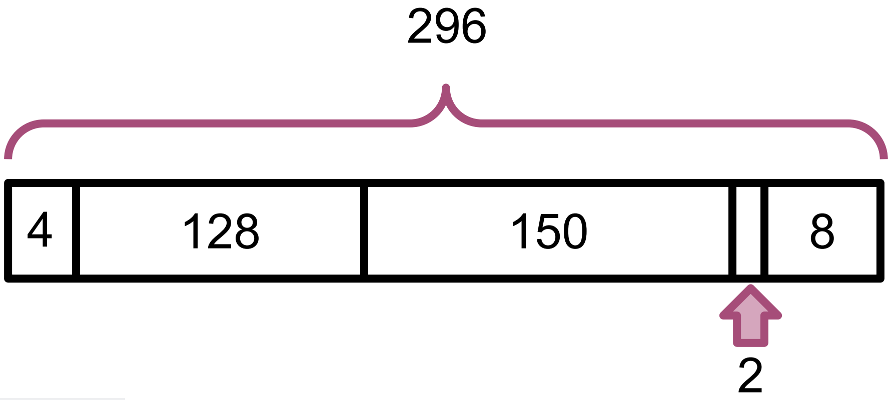
SER 334

Structs

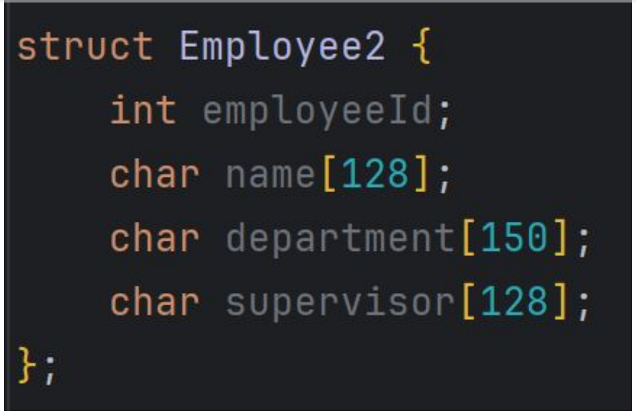
Which structure uses more memory?



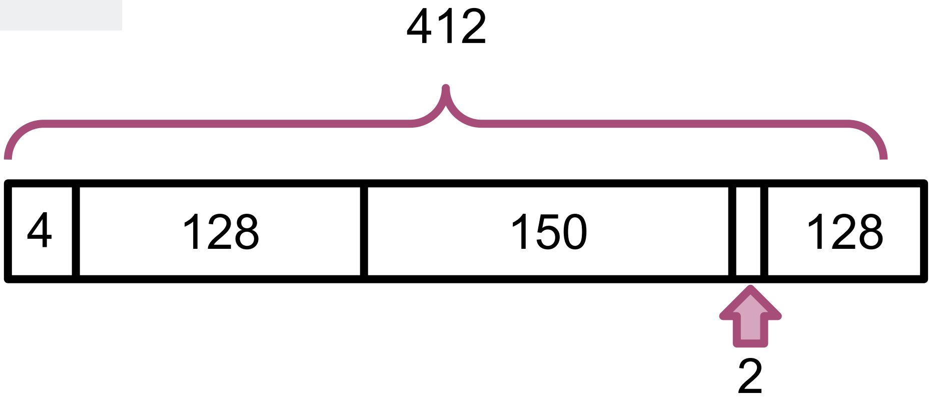
```
struct Employee {
    int employeeId;
    char name[128];
    char department[150];
    Employee *supervisor;
};
```



Check out the recording for a ***GREAT*** *discussion!*

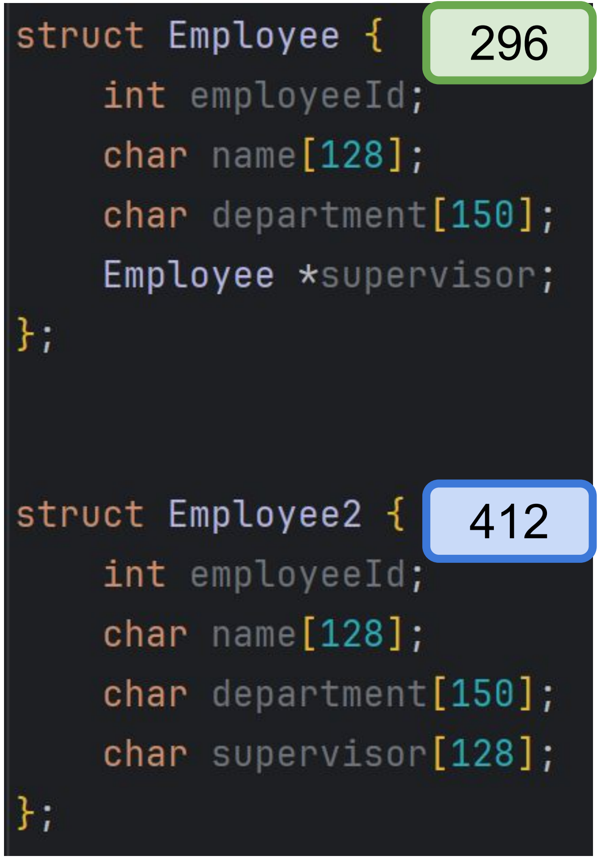


```
struct Employee2 {
    int employeeId;
    char name[128];
    char department[150];
    char supervisor[128];
};
```



SER 334

Structs

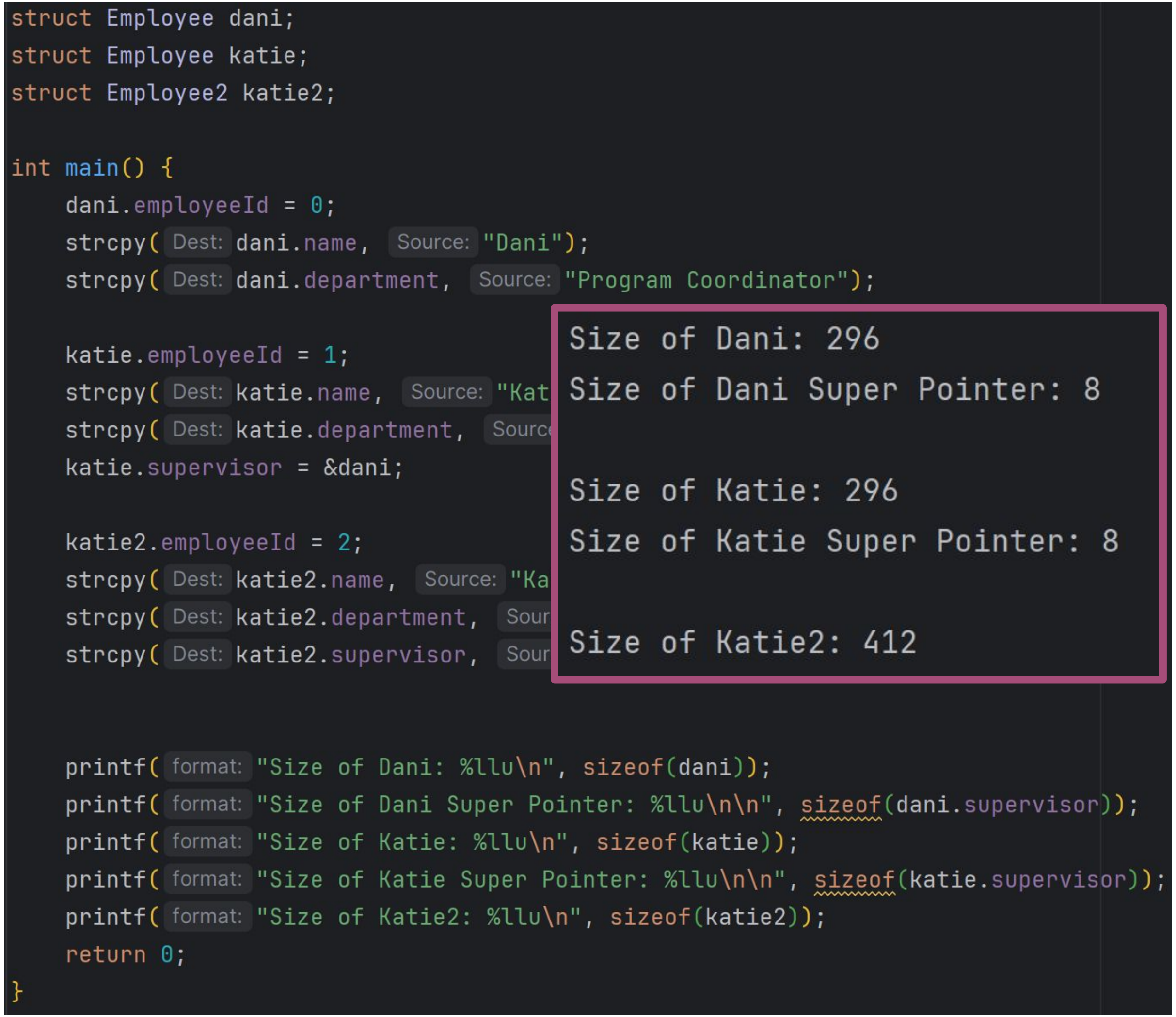


```
struct Employee {
    int employeeId;
    char name[128];
    char department[150];
    Employee *supervisor;
};
```

296

```
struct Employee2 {
    int employeeId;
    char name[128];
    char department[150];
    char supervisor[128];
};
```

412



```
struct Employee dani;
struct Employee katie;
struct Employee2 katie2;

int main() {
    dani.employeeId = 0;
    strcpy( Dest: dani.name,  Source: "Dani");
    strcpy( Dest: dani.department,  Source: "Program Coordinator");

    katie.employeeId = 1;
    strcpy( Dest: katie.name,  Source: "Kat
    strcpy( Dest: katie.department,  Sourc
    katie.supervisor = &dani;

    katie2.employeeId = 2;
    strcpy( Dest: katie2.name,  Source: "Ka
    strcpy( Dest: katie2.department,  Sour
    strcpy( Dest: katie2.supervisor,  Sour


    printf( format: "Size of Dani: %llu\n", sizeof(dani));
    printf( format: "Size of Dani Super Pointer: %llu\n\n", sizeof(dani.supervisor));
    printf( format: "Size of Katie: %llu\n", sizeof(katie));
    printf( format: "Size of Katie Super Pointer: %llu\n\n", sizeof(katie.supervisor));
    printf( format: "Size of Katie2: %llu\n", sizeof(katie2));
    return 0;
}
```

```
Size of Dani: 296
Size of Dani Super Pointer: 8

Size of Katie: 296
Size of Katie Super Pointer: 8

Size of Katie2: 412
```

Upcoming Events

SI Sessions:

- Sunday, January 21st at 7:00 pm MST
- Monday, January 22nd at 7:00 pm MST

- Sunday, January 28th at 7:00 pm MST **Cancelled - good luck on Exam 1!**
- Monday, January 29th at 7:00 pm MST

Review Sessions:

- Exam 1 Review: Thursday, January 25th 7:00 pm - 9:00 pm MST

Questions?

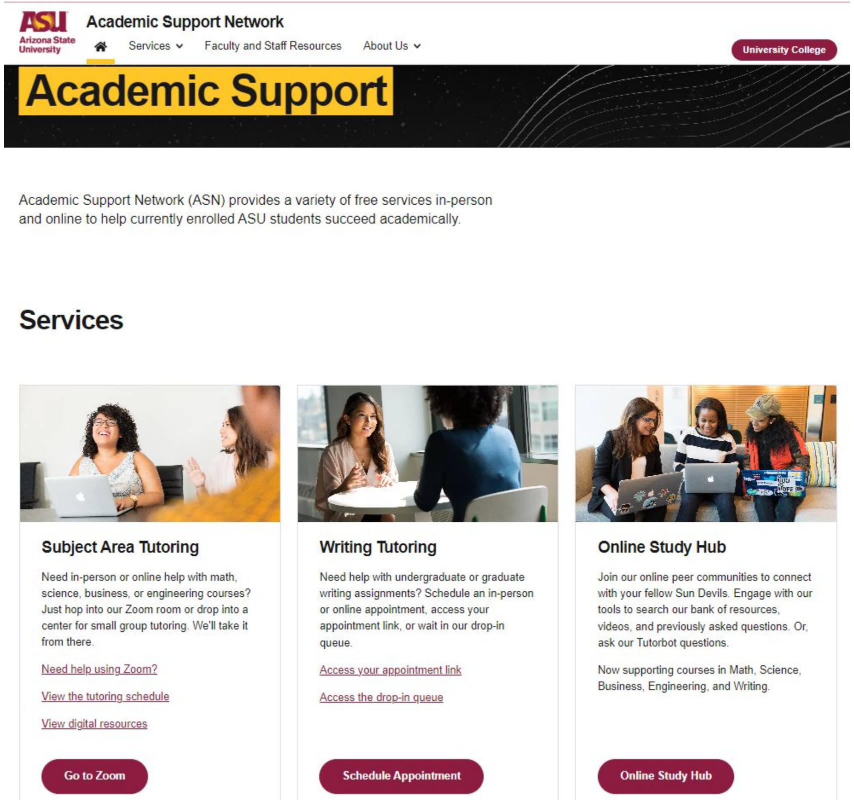
Survey:

http://bit.ly/ASN2324

More Questions?

Check out our other resources!

tutoring.asu.edu



ASU Arizona State University

Academic Support Network

Services ⌄ Faculty and Staff Resources About Us ⌄ University College

Academic Support

Academic Support Network (ASN) provides a variety of free services in-person and online to help currently enrolled ASU students succeed academically.

Services

Subject Area Tutoring

Need in-person or online help with math, science, business, or engineering courses? Just hop into our Zoom room or drop into a center for small group tutoring. We'll take it from there.

Need help using Zoom?

View the tutoring schedule

View digital resources

Go to Zoom

Writing Tutoring

Need help with undergraduate or graduate writing assignments? Schedule an in-person or online appointment, access your appointment link, or wait in our drop-in queue.

Access your appointment link

Access the drop-in queue

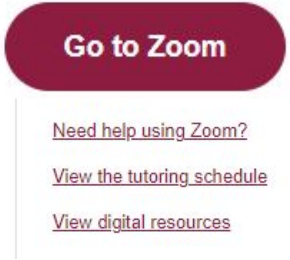
Schedule Appointment

Online Study Hub

Join our online peer communities to connect with your fellow Sun Devils. Engage with our tools to search our bank of resources, videos, and previously asked questions. Or, ask our Tutorbot questions.

Now supporting courses in Math, Science, Business, Engineering, and Writing.

Online Study Hub



1 – Go to Zoom

Need help using Zoom?

2– View the tutoring schedule

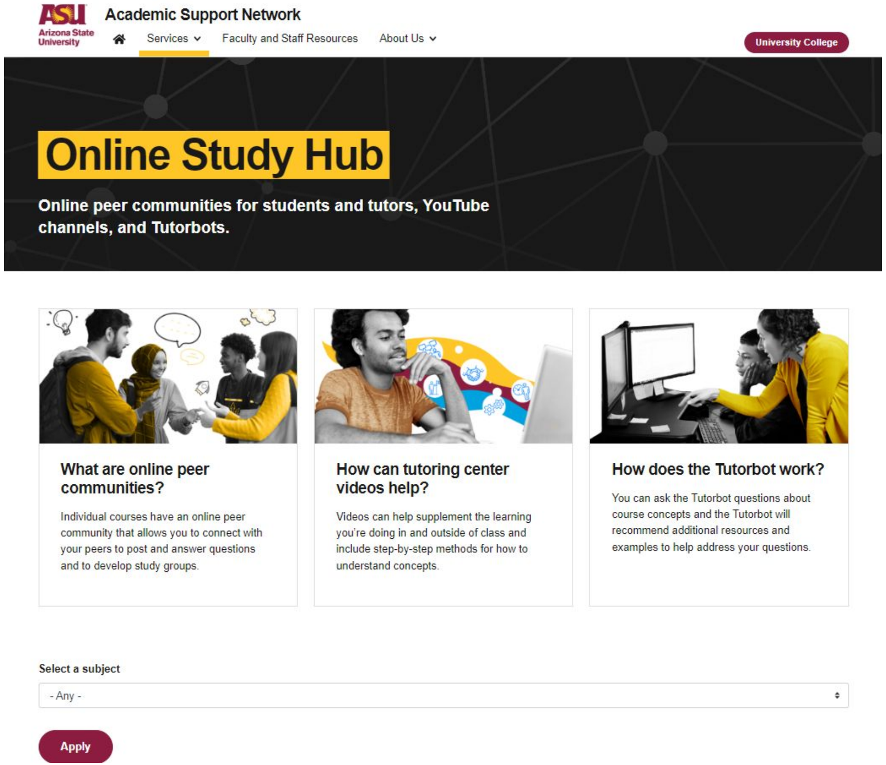
View digital resources

1. Click on ‘Go to Zoom’ to log onto our Online Tutoring Center.
2. Click on ‘View the tutoring schedule’ to see when tutors are available for specific courses.

More Questions?

Check out our other resources!

tutoring.asu.edu/online-study-hub



Academic Support Network

Online Study Hub

Online peer communities for students and tutors, YouTube channels, and Tutorbots.

What are online peer communities?

Individual courses have an online peer community that allows you to connect with your peers to post and answer questions and to develop study groups.

How can tutoring center videos help?

Videos can help supplement the learning you're doing in and outside of class and include step-by-step methods for how to understand concepts.

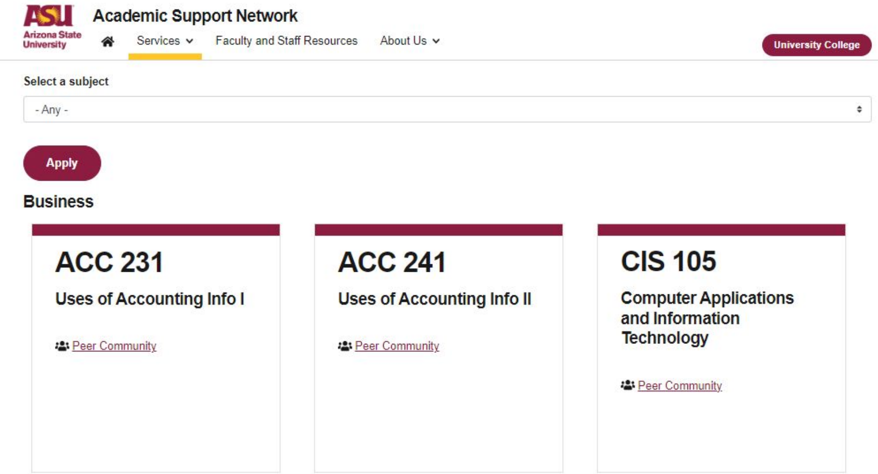
How does the Tutorbot work?

You can ask the Tutorbot questions about course concepts and the Tutorbot will recommend additional resources and examples to help address your questions.

Select a subject

- Any -

Apply



Academic Support Network

Select a subject

- Any -

Apply

Business

ACC 231
Uses of Accounting Info I
Peer Community

ACC 241
Uses of Accounting Info II
Peer Community

CIS 105
Computer Applications and Information Technology
Peer Community

Don't forget to check out
the Online Study Hub
for additional resources!

Additional Resources

- Course Repo
- BMP File Format (Wiki)
- Linux Kernel API