

# Fundamentals of Computing: Lecture 23

Piyush P Kurur  
Office no: 224  
Dept. of Comp. Sci. and Engg.  
IIT Kanpur

September 23, 2009

# Summary of last class

# Summary of last class

1. Allocating memory using the `malloc`,

## Summary of last class

1. Allocating memory using the `malloc`,
2. Freeing memory using `free`,

## Summary of last class

1. Allocating memory using the `malloc`,
2. Freeing memory using `free`,
3. Finding size of a type using `sizeof`,

## Summary of last class

1. Allocating memory using the `malloc`,
2. Freeing memory using `free`,
3. Finding size of a type using `sizeof`,
4. Finding the length of an array `sizeof(a)/sizeof(a[0])`

# Void pointers

# Void pointers

Void pointers are “generic” pointer,



## Void pointers

Void pointers are “generic” pointer,

- ▶ Any pointer can be cast to a void pointer variable

## Void pointers

Void pointers are “generic” pointer,

- ▶ Any pointer can be cast to a void pointer variable

```
void *vptr;  
char *str = "hello";  
vprt = (void *) str;
```

## Void pointers

Void pointers are “generic” pointer,

- ▶ Any pointer can be cast to a void pointer variable

```
void *vptr;  
char *str = "hello";  
vprt = (void *) str;
```

- ▶ If a void pointer variable was assigned a pointer to type T then a pointer to T variable can be cast back from the void pointer.

## Void pointers

Void pointers are “generic” pointer,

- ▶ Any pointer can be cast to a void pointer variable

```
void *vptr;  
char *str = "hello";  
vptr = (void *) str;
```

- ▶ If a void pointer variable was assigned a pointer to type T then a pointer to T variable can be cast back from the void pointer.

```
void *vptr;  
T t, *tptr;  
vptr = (void *) &t;  
...  
tptr = (T *) vptr;
```

## Void pointers

Void pointers are “generic” pointer,

- ▶ Any pointer can be cast to a void pointer variable

```
void *vptr;  
char *str = "hello";  
vptr = (void *) str;
```

- ▶ If a void pointer variable was assigned a pointer to type T then a pointer to T variable can be cast back from the void pointer.

```
void *vptr;  
T t, *tptr;  
vptr = (void *) &t;  
...  
tptr = (T *) vptr;
```

- ▶ A void pointer cannot be dereferenced. Nor can pointer arithmetic be done on it. Reason dereferencing does not make sense and for arithmetic we need to know the size.