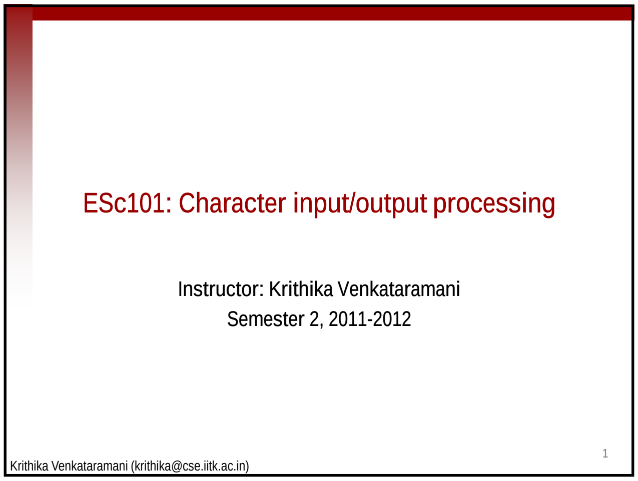
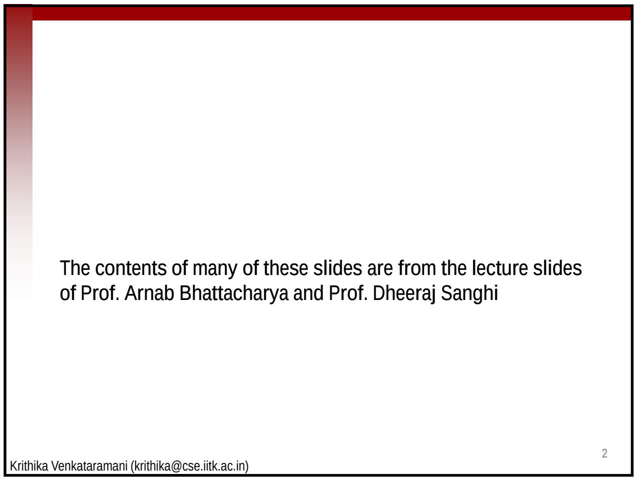
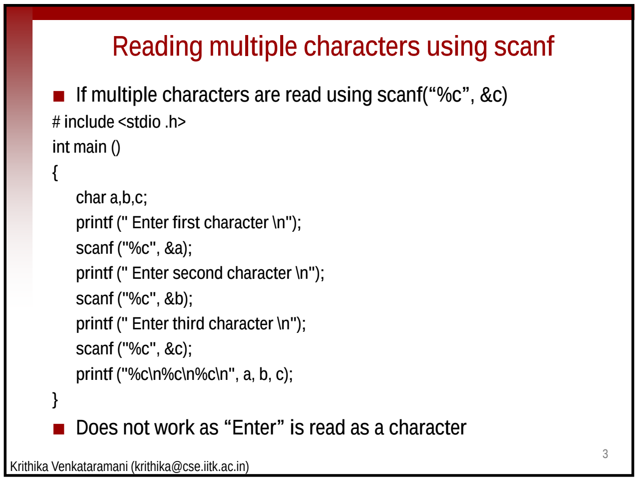


ESc101: Character input/output processing

Instructor: Krithika Venkataraman
Semester 2, 2011-2012



The contents of many of these slides are from the lecture slides of Prof. Arnab Bhattacharya and Prof. Dheeraj Sanghi

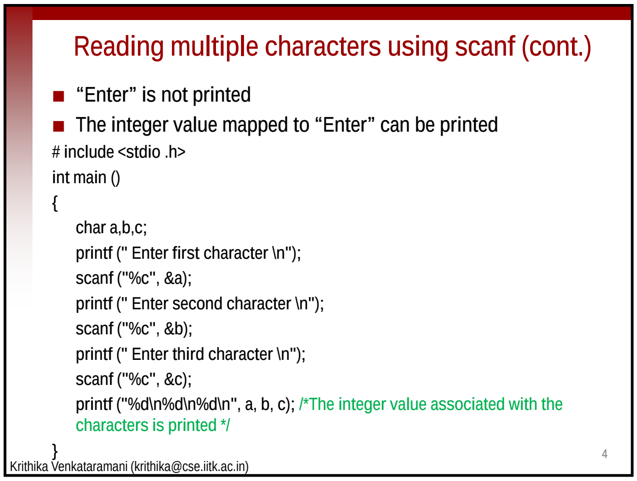


Reading multiple characters using scanf

- If multiple characters are read using scanf(“%c”, &c)

```
# include <stdio .h>
int main ()
{
    char a,b,c;
    printf (" Enter first character \n");
    scanf ("%c", &a);
    printf (" Enter second character \n");
    scanf ("%c", &b);
    printf (" Enter third character \n");
    scanf ("%c", &c);
    printf ("%c\n%c\n%c\n", a, b, c);
}
```

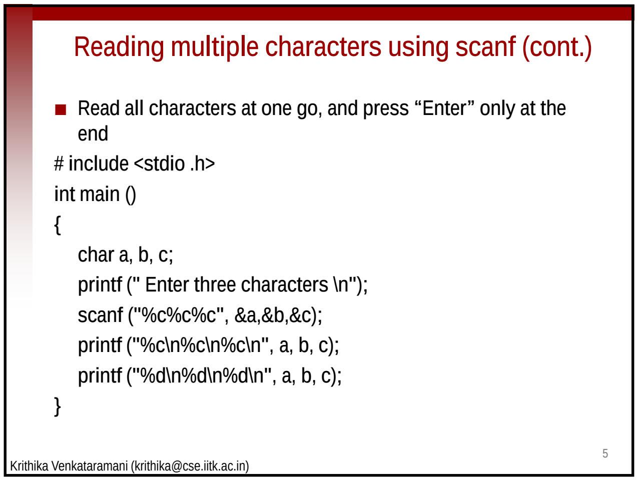
- Does not work as “Enter” is read as a character



Reading multiple characters using scanf (cont.)

- “Enter” is not printed
- The integer value mapped to “Enter” can be printed

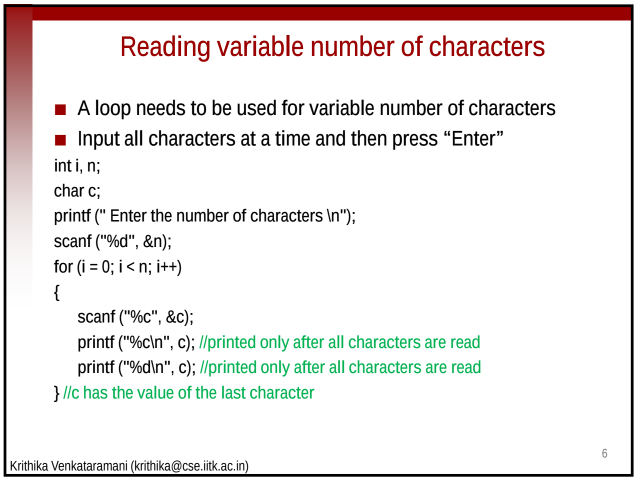
```
# include <stdio .h>
int main ()
{
    char a,b,c;
    printf (" Enter first character \n");
    scanf ("%c", &a);
    printf (" Enter second character \n");
    scanf ("%c", &b);
    printf (" Enter third character \n");
    scanf ("%c", &c);
    printf ("%d\n%d\n%d\n", a, b, c); /*The integer value associated with the
    characters is printed */
}
```



Reading multiple characters using scanf (cont.)

- Read all characters at one go, and press "Enter" only at the end

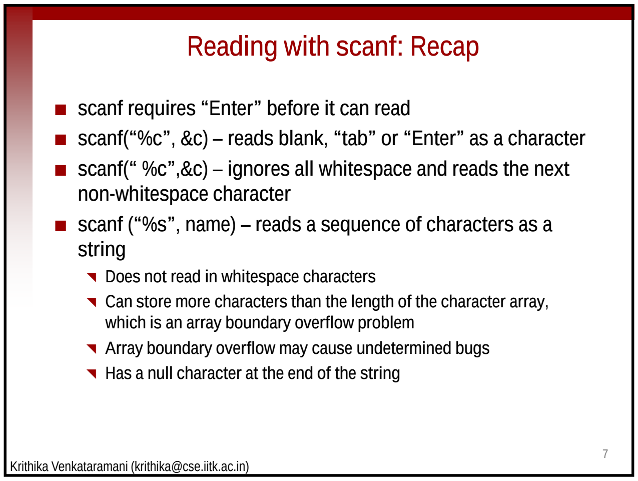
```
# include <stdio .h>
int main ()
{
    char a, b, c;
    printf (" Enter three characters \n");
    scanf ("%c%c%c", &a,&b,&c);
    printf ("%c\n%c\n%c\n", a, b, c);
    printf ("%d\n%d\n%d\n", a, b, c);
}
```



Reading variable number of characters

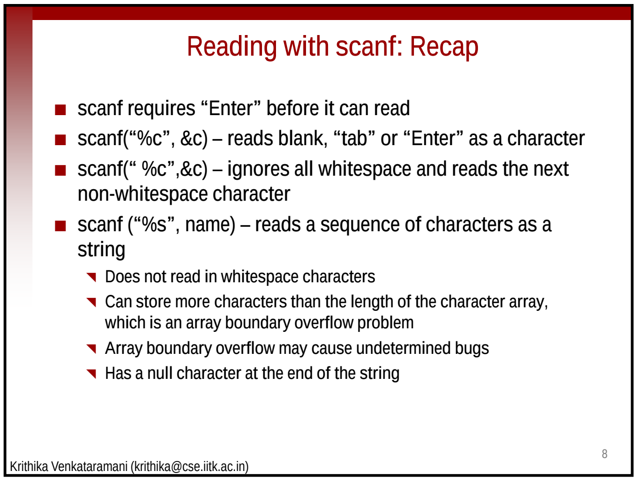
- A loop needs to be used for variable number of characters
- Input all characters at a time and then press "Enter"

```
int i, n;
char c;
printf (" Enter the number of characters \n");
scanf ("%d", &n);
for (i = 0; i < n; i++)
{
    scanf ("%c", &c);
    printf ("%c\n", c); //printed only after all characters are read
    printf ("%d\n", c); //printed only after all characters are read
} //c has the value of the last character
```



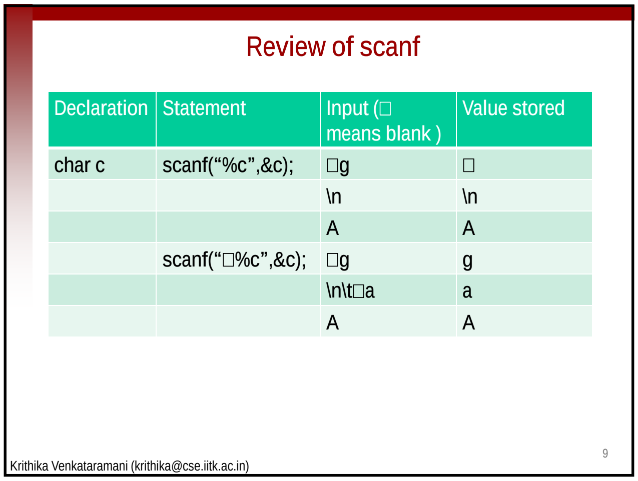
Reading with scanf: Recap

- scanf requires "Enter" before it can read
- scanf("%c", &c) – reads blank, "tab" or "Enter" as a character
- scanf(" %c",&c) – ignores all whitespace and reads the next non-whitespace character
- scanf ("%s", name) – reads a sequence of characters as a string
  - Does not read in whitespace characters
  - Can store more characters than the length of the character array, which is an array boundary overflow problem
  - Array boundary overflow may cause undetermined bugs
  - Has a null character at the end of the string



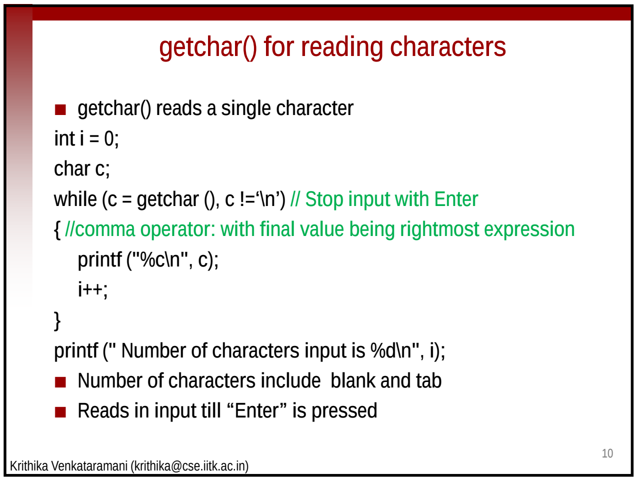
Reading with scanf: Recap

- scanf requires "Enter" before it can read
- scanf("%c", &c) – reads blank, "tab" or "Enter" as a character
- scanf(" %c",&c) – ignores all whitespace and reads the next non-whitespace character
- scanf ("%s", name) – reads a sequence of characters as a string
  - Does not read in whitespace characters
  - Can store more characters than the length of the character array, which is an array boundary overflow problem
  - Array boundary overflow may cause undetermined bugs
  - Has a null character at the end of the string



Review of scanf

Declaration	Statement	Input (□ means blank)	Value stored
char c	scanf(“%c”,&c);	□g	□
		\n	\n
		A	A
	scanf(“□%c”,&c);	□g	g
		\n\t□a	a
		A	A

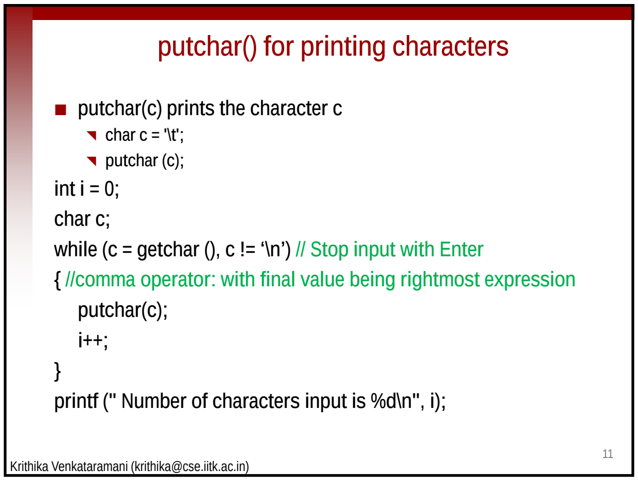


getchar() for reading characters

- getchar() reads a single character

```
int i = 0;
char c;
while (c = getchar (), c !=‘\n’) // Stop input with Enter
{ //comma operator: with final value being rightmost expression
    printf ("%c\n", c);
    i++;
}
printf (" Number of characters input is %d\n", i);
```

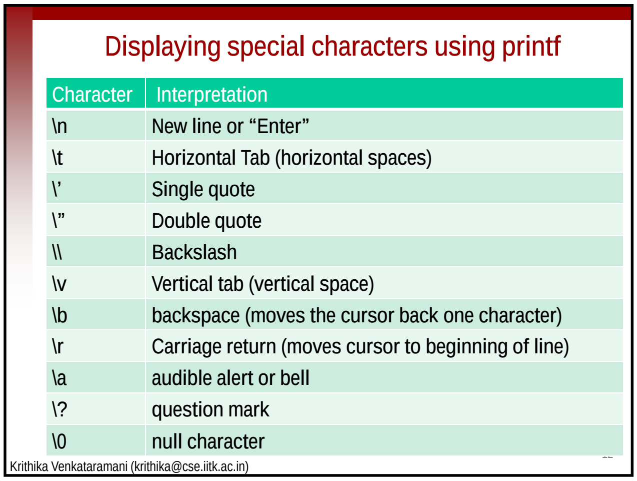
- Number of characters include blank and tab
- Reads in input till “Enter” is pressed



putchar() for printing characters

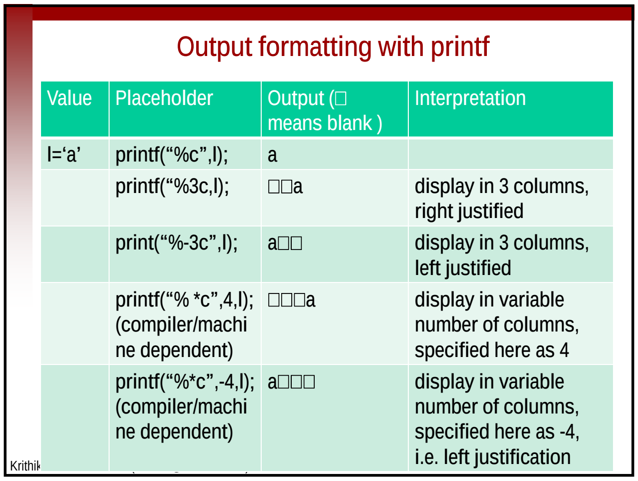
- putchar(c) prints the character c
  - char c = '\t';
  - putchar (c);

```
int i = 0;
char c;
while (c = getchar (), c != ‘\n’) // Stop input with Enter
{ //comma operator: with final value being rightmost expression
    putchar(c);
    i++;
}
printf (" Number of characters input is %d\n", i);
```



Displaying special characters using printf

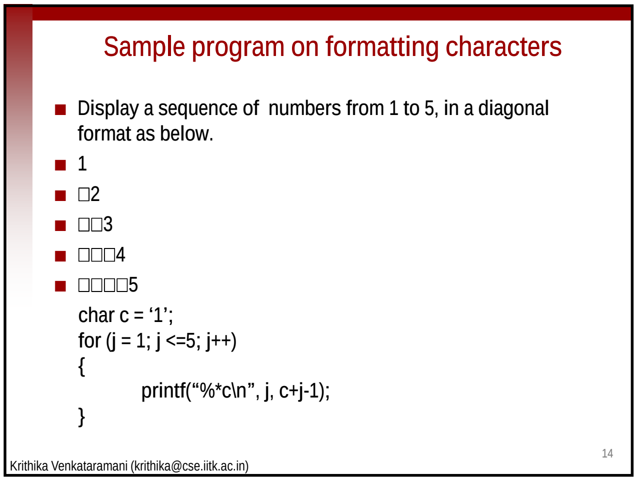
Character	Interpretation
\n	New line or “Enter”
\t	Horizontal Tab (horizontal spaces)
\’	Single quote
\”	Double quote
\\	Backslash
\v	Vertical tab (vertical space)
\b	backspace (moves the cursor back one character)
\r	Carriage return (moves cursor to beginning of line)
\a	audible alert or bell
\?	question mark
\0	null character



Output formatting with printf

Value	Placeholder	Output (□ means blank)	Interpretation
l=‘a’	printf(“%c”,l);	a	
	printf(“%3c,l);	□□a	display in 3 columns, right justified
	print(“%-3c”,l);	a□□	display in 3 columns, left justified
	printf(“% *c”,4,l); (compiler/machine dependent)	□□□a	display in variable number of columns, specified here as 4
	printf(“%*c”,-4,l); (compiler/machine dependent)	a□□□	display in variable number of columns, specified here as -4, i.e. left justification

Krithik



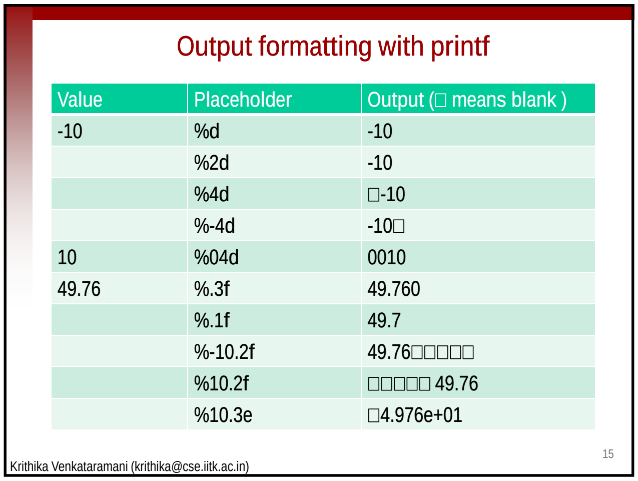
Sample program on formatting characters

- Display a sequence of numbers from 1 to 5, in a diagonal format as below.
- 1
- □2
- □□3
- □□□4
- □□□□5

```
char c = ‘1’;
for (j = 1; j <=5; j++)
{
        printf(“%*c\n”, j, c+j-1);
}
```

14

Krithika Venkataramani (krithika@cse.iitk.ac.in)

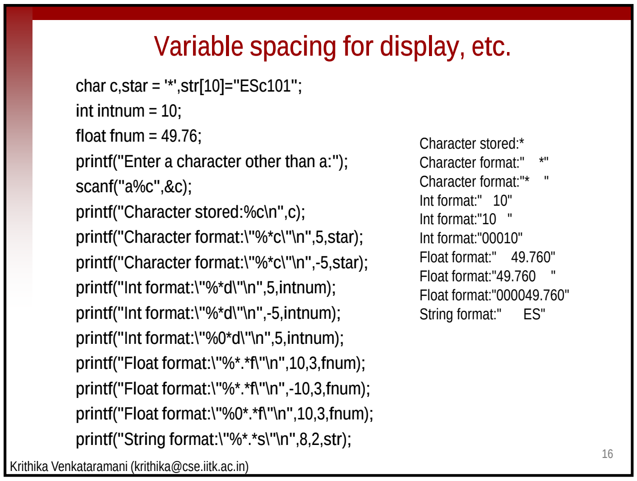


Output formatting with printf

Value	Placeholder	Output (□ means blank)
-10	%d	-10
	%2d	-10
	%4d	□-10
	%-4d	-10□
10	%04d	0010
49.76	%.3f	49.760
	%.1f	49.7
	%-10.2f	49.76□□□□□
	%10.2f	□□□□□ 49.76
	%10.3e	□4.976e+01

15

Krithika Venkataramani (krithika@cse.iitk.ac.in)



Variable spacing for display, etc.

```
char c,star = '*',str[10]="ESc101";
int intnum = 10;
float fnum = 49.76;
printf("Enter a character other than a:");
scanf("a%c",&c);
printf("Character stored:%c\n",c);
printf("Character format:\"%*c\"\n",5,star);
printf("Character format:\"%*c\"\n",-5,star);
printf("Int format:\"%*d\"\n",5,intnum);
printf("Int format:\"%*d\"\n",-5,intnum);
printf("Int format:\"%0*d\"\n",5,intnum);
printf("Float format:\"%*.*f\"\n",10,3,fnum);
printf("Float format:\"%*.*f\"\n",-10,3,fnum);
printf("Float format:\"%0*.*f\"\n",10,3,fnum);
printf("String format:\"%*.*s\"\n",8,2,str);
```

```
Character stored:*
Character format:"    *"
Character format:"*    "
Int format:"   10"
Int format:"10   "
Int format:"00010"
Float format:"    49.760"
Float format:"49.760    "
Float format:"000049.760"
String format:"      ES"
```

16

Krithika Venkataramani (krithika@cse.iitk.ac.in)