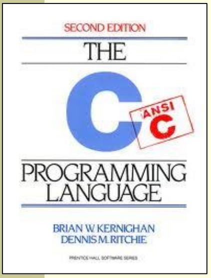
CS 241
Data Organization using C
If statements

Instructor: **Joel Castellanos**
e-mail: joel@unm.edu
Web: http://cs.unm.edu/~joel/
Office: Farris Engineering Center, Room 2110

9/3/2019

Read: Kernighan & Ritchie



- Due Tuesday, Sept 3
  - 1.10: External Variables and Scope
  - 2.1: Variable Names
  - 2.2: Data Types and Sizes
- Due Thursday, Sept 5
  - 2.3: Constants
  - 2.4: Declarations
  - 2.5: Arithmetic Operators
  - 2.6: Relational and Logical Operators
  - 2.7: Type Conversions
  - 2.8: Increment and Decrement Operators
  - 2.9: Bitwise Operations

IF, ELSE IF, and ELSE

```c
#include <stdio.h>
void main(void)
{ char c = getchar();
  if (c == 'c')
  { printf("Club\n");
  }
  else if (c == 'd')
  { printf("Diamond\n");
  }
  else if (c == 'h')
  { printf("Hart\n");
  }
  else
  { printf("Spade\n");
  }
}
```

Read one character from the standard input stream (keyboard).

What does this program do?

What is a likely logic error?

What is Wrong?

```c
#include <stdio.h>
void main(void)
{
  int grade = 87;

  if (grade >= 90);
  { printf("You get an A\n");
  }

  if (grade < 60);
  { printf("You Fail\n");
  }
}
```

Output:

```
You get an A
You Fail
```

What is Wrong?

```
1)  #include <stdio.h>
2)  void main(void)
3)  {
4)    int grade = 87;
5)
6)    if (grade >= 90)
7)      printf("Congratulations\n");
8)      printf("You get an A\n");
9)
10)
11)   if (grade < 60)
12)   { printf("You Fail\n");
13)   }
14) }
```

Output:
```
You get an A
```

5

5

How Can This Logic Be Improved?

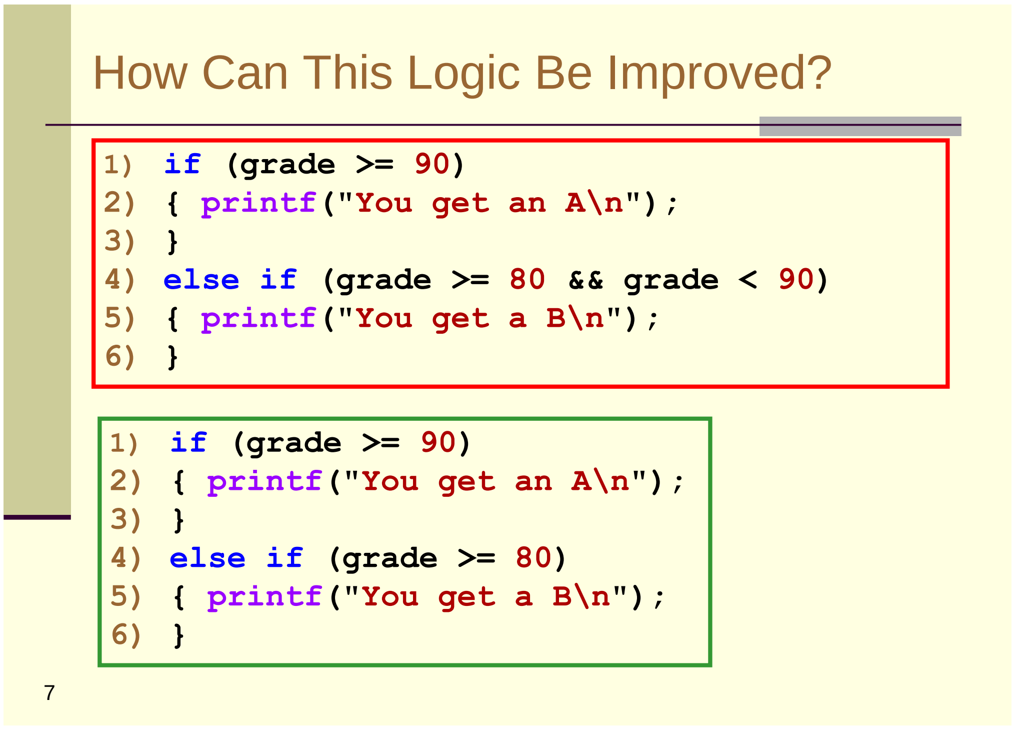
```
1)  if (grade >= 90)
2)  { printf("You get an A\n");
3)  }
4)
5)  if (grade < 60)
6)  { printf("You Fail\n");
7)  }
```

```
1)  if (grade >= 90)
2)  { printf("You get an A\n");
3)  }
4)
5)  else if (grade < 60)
6)  { printf("You Fail\n");
7)  }
```

Since it is impossible for ***both*** (1) & (5) to be true.

6

6



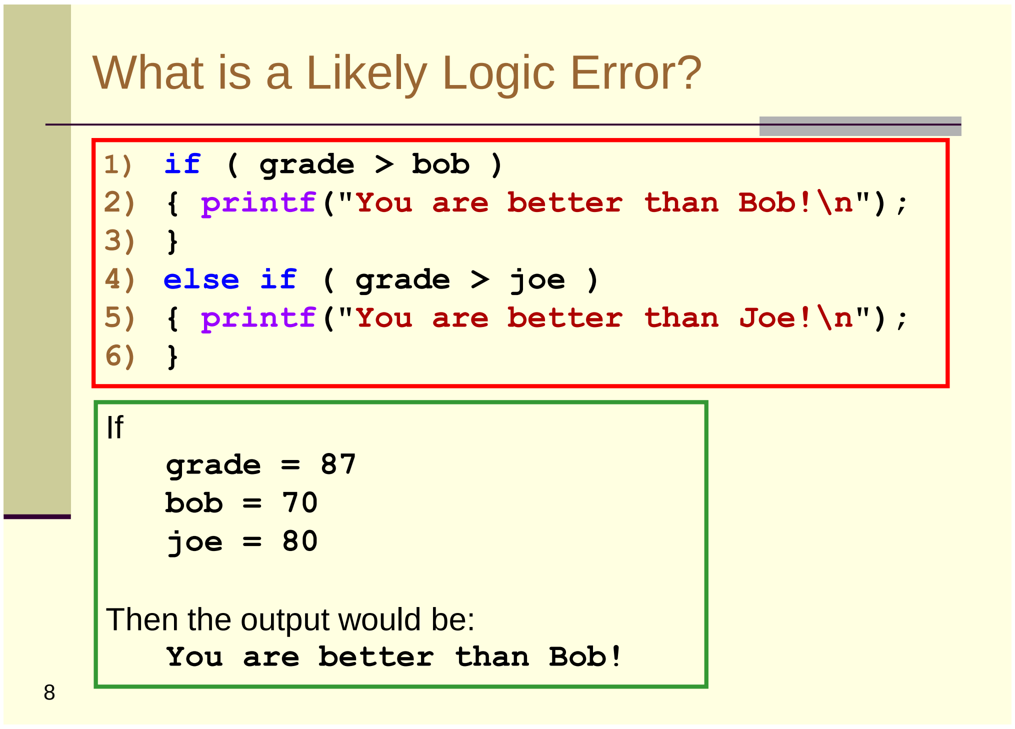
How Can This Logic Be Improved?

```
if (grade >= 90)
{ printf("You get an A\n");
}
else if (grade >= 80 && grade < 90)
{ printf("You get a B\n");
}
```

```
if (grade >= 90)
{ printf("You get an A\n");
}
else if (grade >= 80)
{ printf("You get a B\n");
}
```

7

7



What is a Likely Logic Error?

```
if ( grade > bob )
{ printf("You are better than Bob!\n");
}
else if ( grade > joe )
{ printf("You are better than Joe!\n");
}
```

If

```
grade = 87
bob = 70
joe = 80
```

Then the output would be:

```
You are better than Bob!
```

8

8

IF, ELSE IF, ELSE

```
void main(void)
{
  int x = 1;

  if (x == 1)          ¡¡¡No semicolon!!!
  { printf("x is 1\n");
  }
  else if (x == 2)     ¡¡¡No semicolon!!!
  { printf("x is 2\n");
  }
  else                 ¡¡¡No semicolon!!!
  { printf("x is special\n");
  }
}
```

9

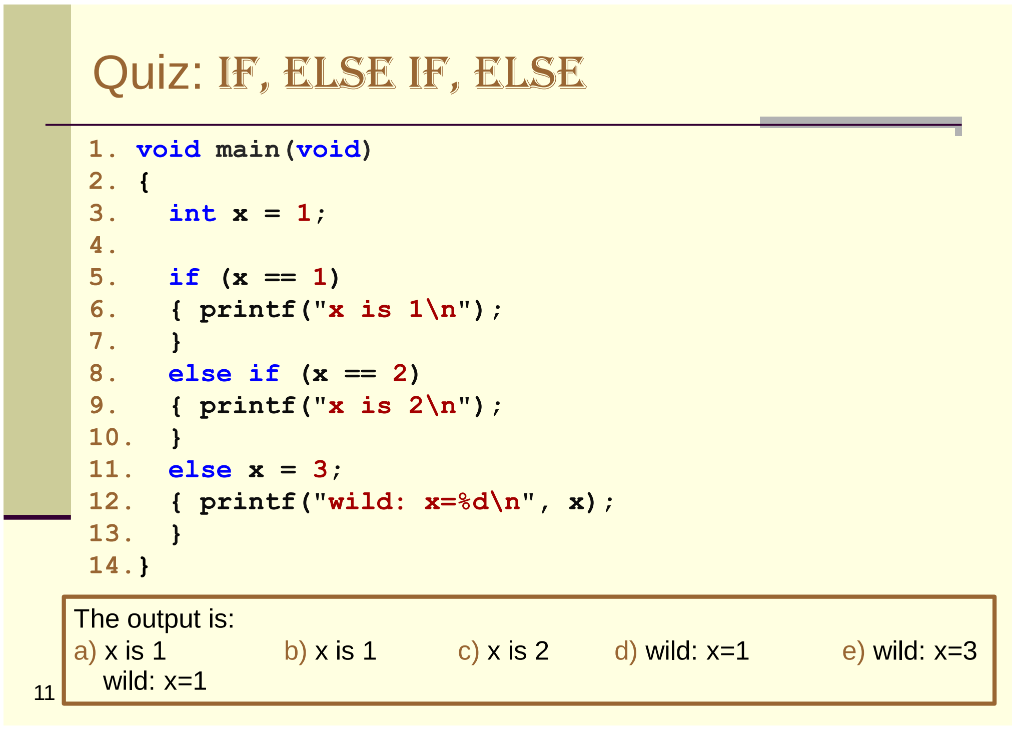
IF, ELSE IF, ELSE

```
void main(void)
{ int x = 5;
  int y = 4;
  if (x + y > 10)
  { printf("Here 1\n");
    if (x + y > 2)
    printf("Here 2\n");
  }
  else if (x + y > 7)
  { printf("Here 3\n");
  }
  else if (x+y < 15)
  { printf("Here 4\n");
    if (x+y > 1) printf("Here 6\n");
  }
  printf("Here 5\n");
}
```

Output:

```
Here 3
Here 5
```

10



Quiz: IF, ELSE IF, ELSE

```
void main(void)
{
  int x = 1;

  if (x == 1)
  { printf("x is 1\n");
  }
  else if (x == 2)
  { printf("x is 2\n");
  }
  else x = 3;
  { printf("wild: x=%d\n", x);
  }
}
```

The output is:

a) x is 1
   wild: x=1

b) x is 1

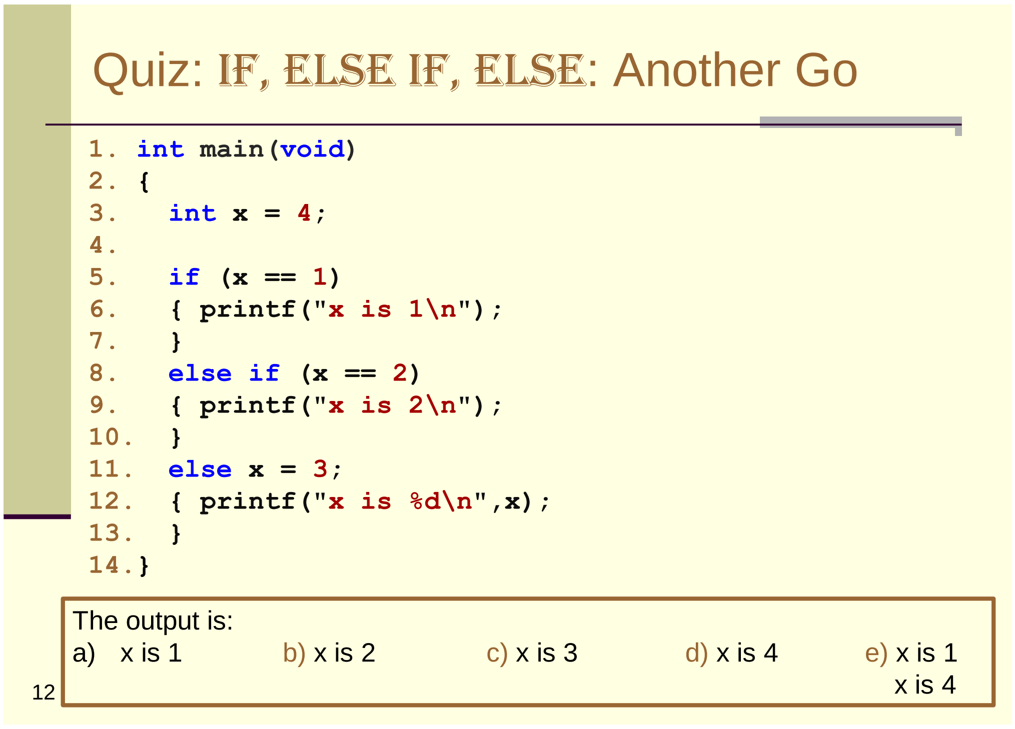
c) x is 2

d) wild: x=1

e) wild: x=3

11

11



Quiz: IF, ELSE IF, ELSE: Another Go

```
int main(void)
{
  int x = 4;

  if (x == 1)
  { printf("x is 1\n");
  }
  else if (x == 2)
  { printf("x is 2\n");
  }
  else x = 3;
  { printf("x is %d\n",x);
  }
}
```

The output is:

a) x is 1

b) x is 2

c) x is 3

d) x is 4

e) x is 1
   x is 4

12

12

charSequenceToInt.c 1 of 2

```c
#include <stdio.h>

int main(void)
{ int number = 0;
  char c='\n';

  while  (1)
  { c = getchar();

    //Return when End Of File is reached
    //Return if error is found
    //Process the input data

  } //end of while (1)
} //end of main()
```

13

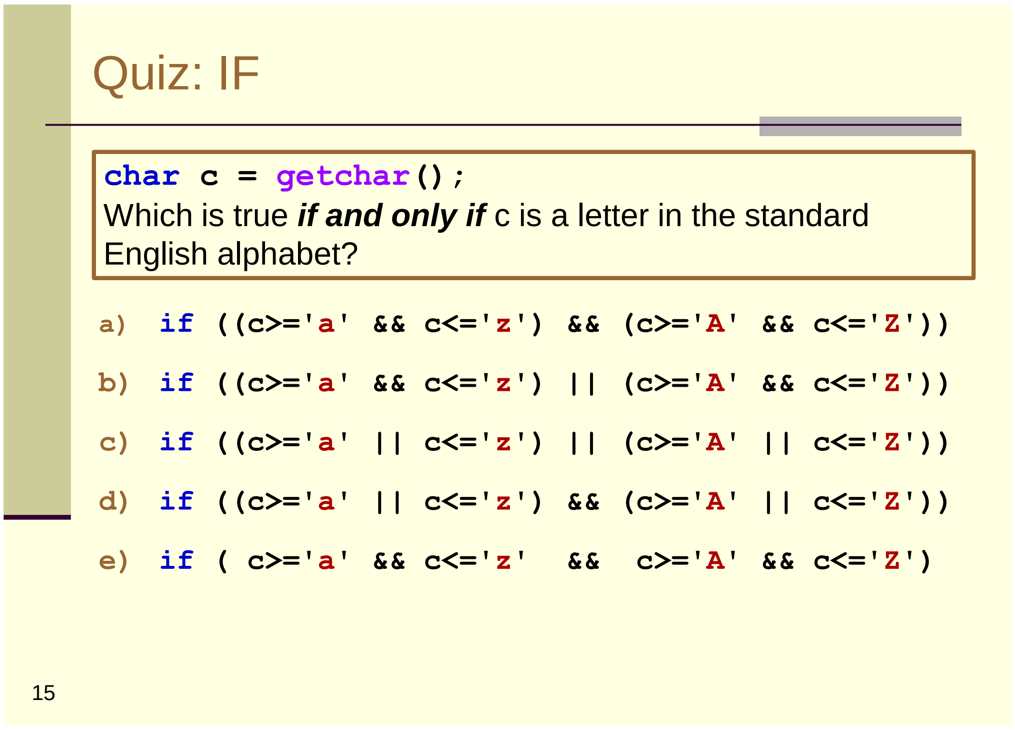
13

charSequenceToInt.c 2 of 2

```c
while  (1)
{ c = getchar();
  if (c == EOF) return 0;
  if (c >= '0' && c <= '9')
  { if (number > 32000 )
    { printf("Error** exceeds max value 32000\n");
      return -1; //error code
    }
    number = number*10 + c-'0';
  }
  else if (c == '\n')
  { printf("The number is: %d\n", number);
    number=0;
  }
  else
  { printf("Error** Not digit 0 - 9: <%c>\n", c);
    return -1; //error code
  }
} //end of:  while (1)
```

14

14



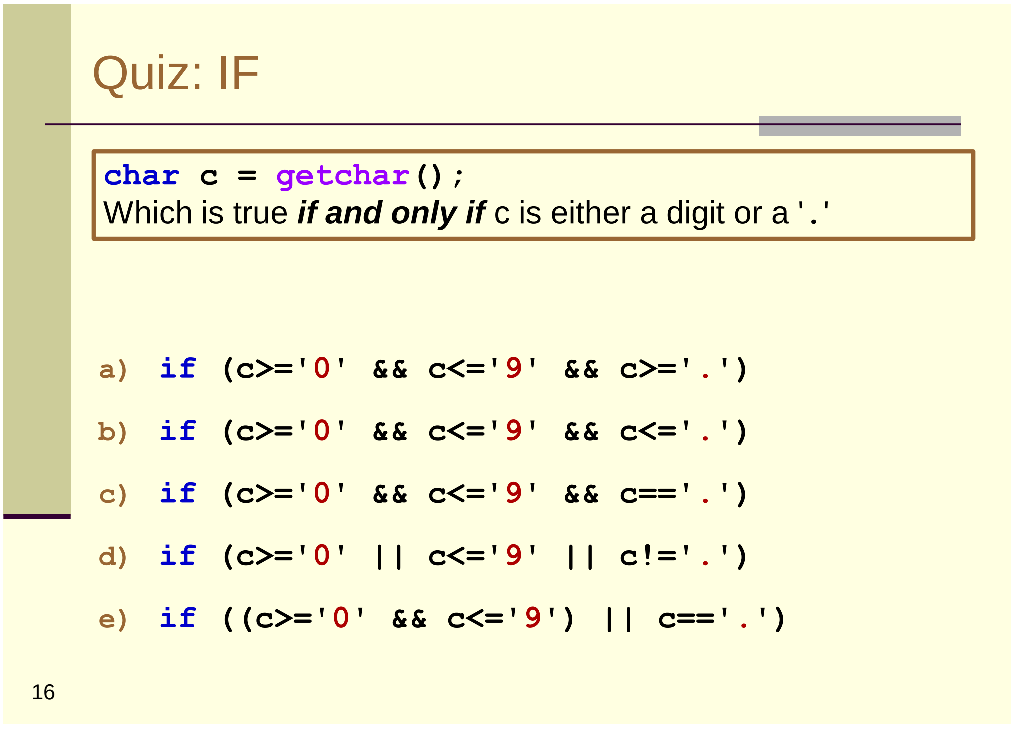
Quiz: IF

```
char c = getchar();
```
Which is true ***if and only if*** c is a letter in the standard English alphabet?

```
a)  if ((c>='a' && c<='z') && (c>='A' && c<='Z'))
b)  if ((c>='a' && c<='z') || (c>='A' && c<='Z'))
c)  if ((c>='a' || c<='z') || (c>='A' || c<='Z'))
d)  if ((c>='a' || c<='z') && (c>='A' || c<='Z'))
e)  if ( c>='a' && c<='z'  &&  c>='A' && c<='Z')
```

15

15



Quiz: IF

```
char c = getchar();
```
Which is true ***if and only if*** c is either a digit or a `'.'`

```
a)  if (c>='0' && c<='9' && c>='.')
b)  if (c>='0' && c<='9' && c<='.')
c)  if (c>='0' && c<='9' && c=='.')
d)  if (c>='0' || c<='9' || c!='.')
e)  if ((c>='0' && c<='9') || c=='.')
```

16

16

Using the Modulus Operator: %

```
int i;
for (i=12; i>=0; i--)
{
  printf("%d ", i%5);
}
printf("\n");

for (i=4; i>=0; i--)
{
  printf("%d ", 5%i);
}
```

Output

```
2 1 0 4 3 2 1 0 4 3 2 1 0
Floating exception
```

17

Flag

In computer programming, ***flag*** often refers to a variable or bit used to indicate a particular property is "*on*" or "*off* ".

18

Quiz: What is the Output?

```c
int i, n;
int flag;
for (n=10; n>1; n--)
{ flag = 0;
  for (i=2; i<n; i++)
  { if (n % i == 0) flag = 1;
  }
  if (flag == 0) printf("%d ", n);
}
printf("\n");
```

a) 10 9 8 7 6 5 4 3 2
b) 9 8 7 6 5 4 3
c) 9 7 5 3
d) 7 5 3 2

More Efficient Printer of Primes

```c
int i, n;
int flag;
for (n=10; n>1; n--)
{ flag = 0;
  for (i=2; i<n; i++)
  { if (n % i == 0)
    { flag = 1;
      break;
    }
  }
  if (flag == 0) printf("%d ", n);
}
printf("\n");
```

When one factor of n is found, n is cannot be prime, so break out of the inner loop.

Interview Question

- Describe a method to determine if a given number is prime? What is its time complexity?
- What are the most important things that could be done to make your prime determination method efficient? What is its time complexity?
- What is an O(*n*) (linear time) method to print all primes up to a given number *n*? Hint: the only method I know of also takes linear space.