APS105
Winter 2012

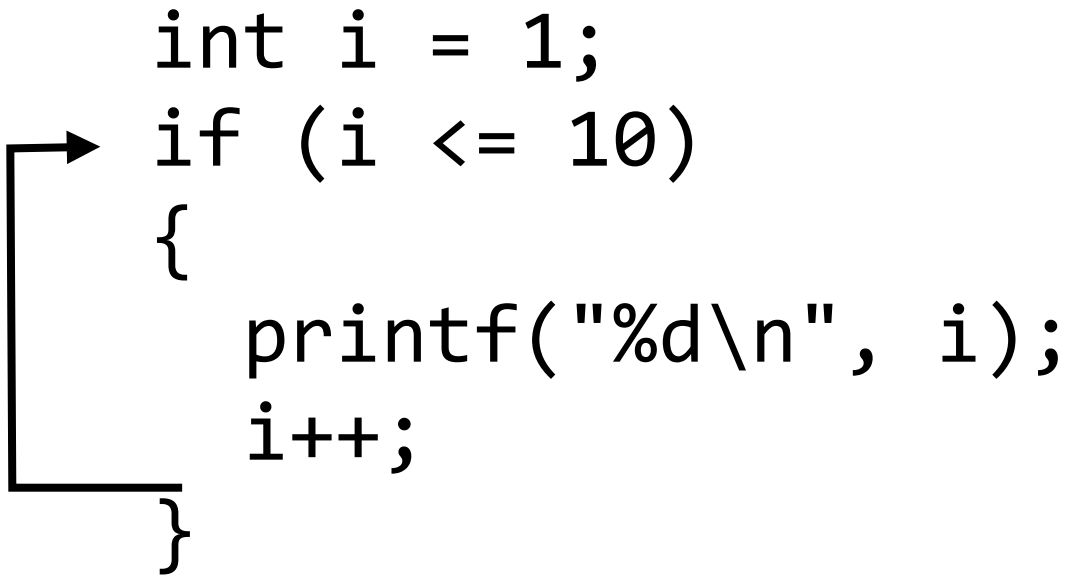
Jonathan Deber
jdeber -at- cs -dot- toronto -dot- edu

Lecture 8
February 1, 2012

Today

- `for` loops
- Loop idioms and categories
- Loop examples

```
int i = 1;
while (i <= 10)
{
   printf("%d\n", i);
   i++;
}
```



for loop

```
for (expression1; expression2; expression3)
    statement

for (initializer; condition; increment)
    statement


initializer
while (condition)
{
    statement
    increment
}
```

```
for (int i = 1; i <= 10; i++)
{
    printf("%d\n", i);
}
```

```
int i = 1;
while (i <= 10)
{
    printf("%d\n", i);
    i++;
}
```

```
for ( ; i != 5; )
{
   // do something
}
```

```
while (i != 5)
{
   // do something
}
```

```
for ( ; ; )
{
   // do something
}
```

```
while (true)
{
   // do something
}
```

Multivariable Loops

```
int i, j;

for (i = 0, j = 0; i < 5; i++, j++)
{
    printf("%d x %d is %d \n", i, j, i * j);
}
```

```
0 x 0 is 0
1 x 1 is 1
2 x 2 is 4
3 x 3 is 9
4 x 4 is 16
```

```
for (int i = 0; i < 10; i++)
{
    ...
}
int maximum = i;
```

Error

```
int i;
for (i = 0; i < 10; i++)
{
    ...
}
int maximum = i;
```

Loop Idiom: Variable Naming

- i is a counter variable

```
for (int i = 0; i < 10; i++)
{
    ...
}

for (int index = 0; index < 10; index++)
{
    ...
}
```

Loop Idiom: n times

- Let's say you want to something n times

n: 3

```
for (i = 0; i < n; i++)
```
0 1 2

```
for (i = 1; i <= n; i++)
```
1 2 3

```
for (i = 0; i <= n; i++)
```
0 1 2 3

Wrong

Loop Idiom: Alternating

```
bool printAsterisk = true;
for (int i = 0; i < num; i++)
{
    if (printAsterisk)
    {
        printf("*");
    }
    else
    {
        printf("-");
    }
    printAsterisk = !printAsterisk;
}
```

num: 8

```
*-*-*-*-
```

Loop Idiom: Alternating

```
bool printAsterisk = true;
for (int i = 0; i < num; i++)
{
    char c = printAsterisk ? '*' : '-';

    printf("%c", c);

    printAsterisk = !printAsterisk;
}
```

num: 8

```
*-*-*-*-
```

Categories of Loops

- Read and process
- Validity checking
- Counting loop

Read and Process

```
int num;
while (true)
{
   printf("Enter a number: ");
   scanf("%d", &num);

   int square = num * num;
   printf("%d squared is %d \n", num, square);
}
```

```
bool keepGoing = true;
while (keepGoing)
{
  int num;
  printf("Enter a number (0 to quit): ");
  scanf("%d", &num);

  if (num == 0)
  {
    keepGoing = false;
  }
  else
  {
    int square = num * num;
    printf("%d squared is %d \n", num, square);
  }
}
```

Validity Checking

```
int num;
printf("Enter a positive number: ");
scanf("%d", &num);

while (num < 0)
{
   printf("Enter a positive number: ");
   scanf("%d", &num);
}

// At this point, num must be positive
```

Counting

```
for (int i = 0; i < num; i++)
{
    printf("%d\n", i);
}
```

How to Choose a Loop

```
for (int i = 0; i < 10; i++)
{
    printf("%d\n", i);
}
```

```
int i = 0
while (i < 10)
{
    printf("%d\n", i);
    i++;
}
```

How to Choose a Loop

- `for` loop
  - Well-defined number of iterations
  - Counting loops
- `while` loop
  - Unknown number of iterations
  - Validity checking
  - Read and process

Nested ifs

```
if ( (num % 2) == 0 )
{
   printf("%d is even\n", num);

   if ( (num % 3) == 0 )
   {
      printf("it's divisible by 3\n");
   }
}
```

Nested Loops

```
int num;
while (true)
{
  printf("Enter a number:");
  scanf("%d", &num);

  // do something
}
```

Nested Loops

```
int num;
while (true)
{
   printf("Enter a number:");
   scanf("%d", &num);


}
```

Nested Loops

```
int num;
while (true)
{
   printf("Enter a number:");
   scanf("%d", &num);

   int sum = 0;
   for (int i = 1; i <= num; i++)
   {
      sum += i;
   }
   printf("The sum of 1 to %d is %d", num, sum);

}
```

Nested Loops

```
int num;
while (true)
{
   printf("Enter a number:");
   scanf("%d", &num);

   int sum = 0;
   for (int i = 1; i <= num; i++)
   {
      sum += i;
   }
   printf("The sum of 1 to %d is %d", num, sum);

}
```

Outer loop

Inner loop

```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```

X_MAX: 3

Y_MAX: 2

```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```

X_MAX: 3 x: 0

Y_MAX: 2

```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```

X_MAX: 3 x: 0 y: 0

Y_MAX: 2

```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```

X_MAX: 3

Y_MAX: 2

x: 0

y: ~~0~~ 1

```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```

X_MAX: 3

Y_MAX: 2

x: 0

y: ~~0~~ ~~1~~ 2

```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```

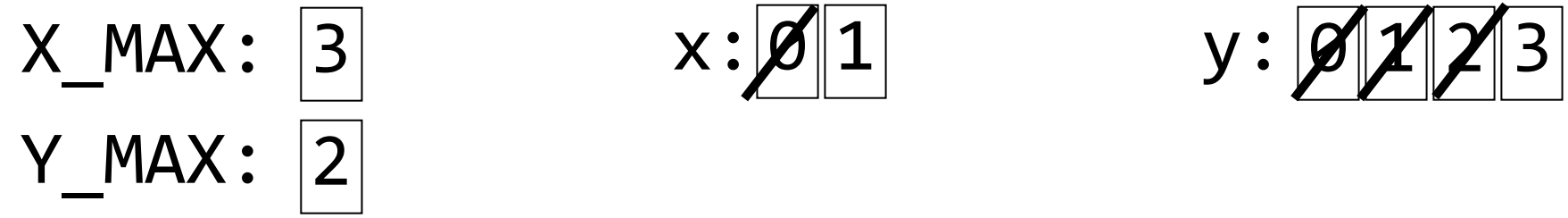
X_MAX: 3

Y_MAX: 2

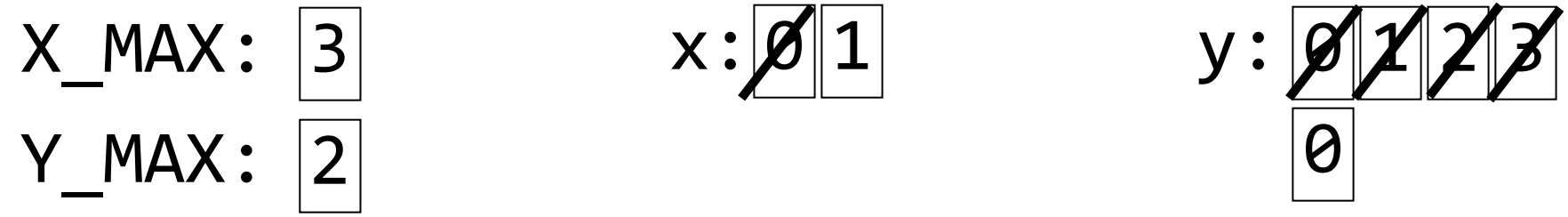
x: 0

y: ~~0~~ ~~1~~ ~~2~~ 3

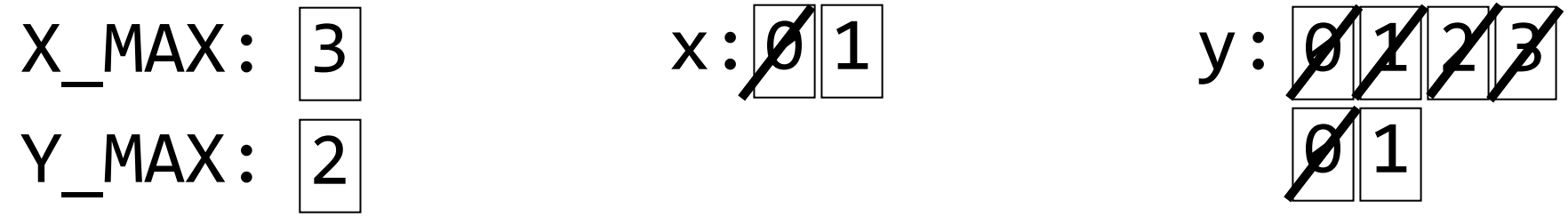
```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```



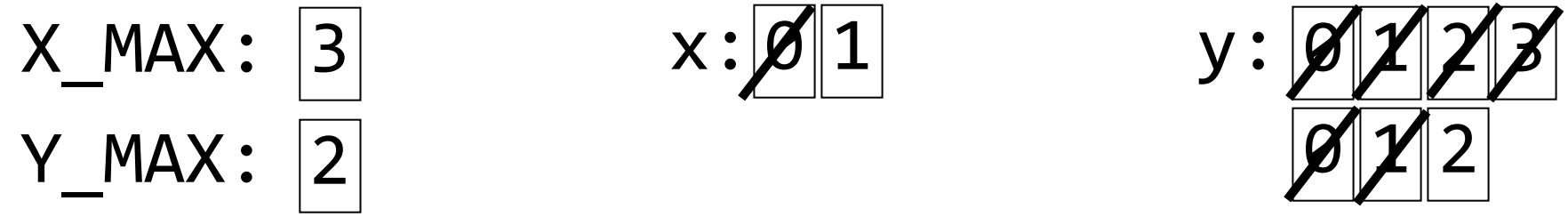
```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```



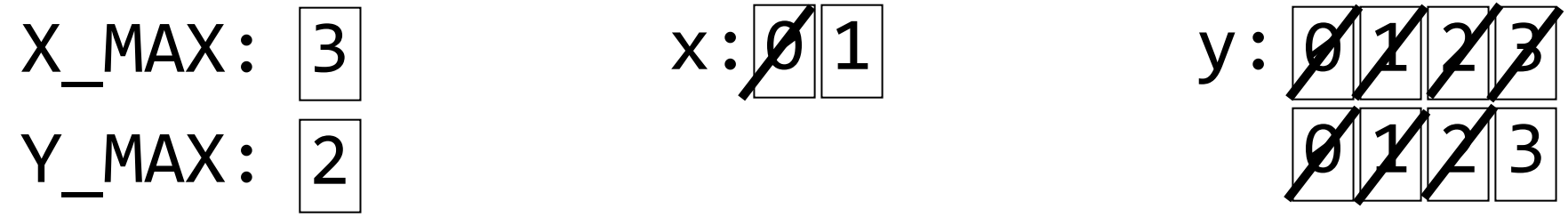
```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```



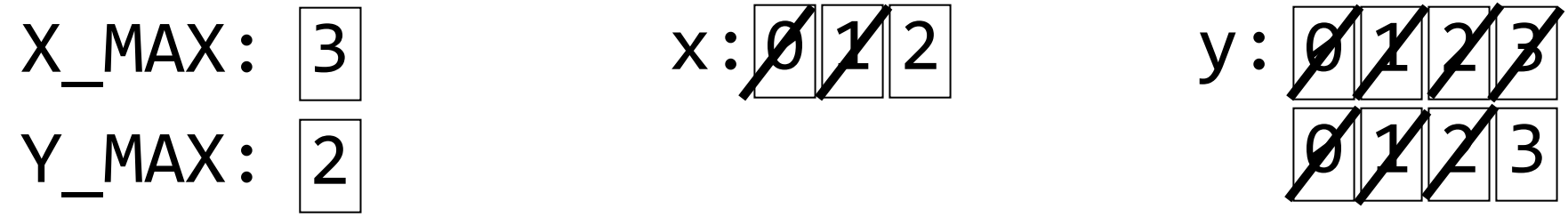
```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```



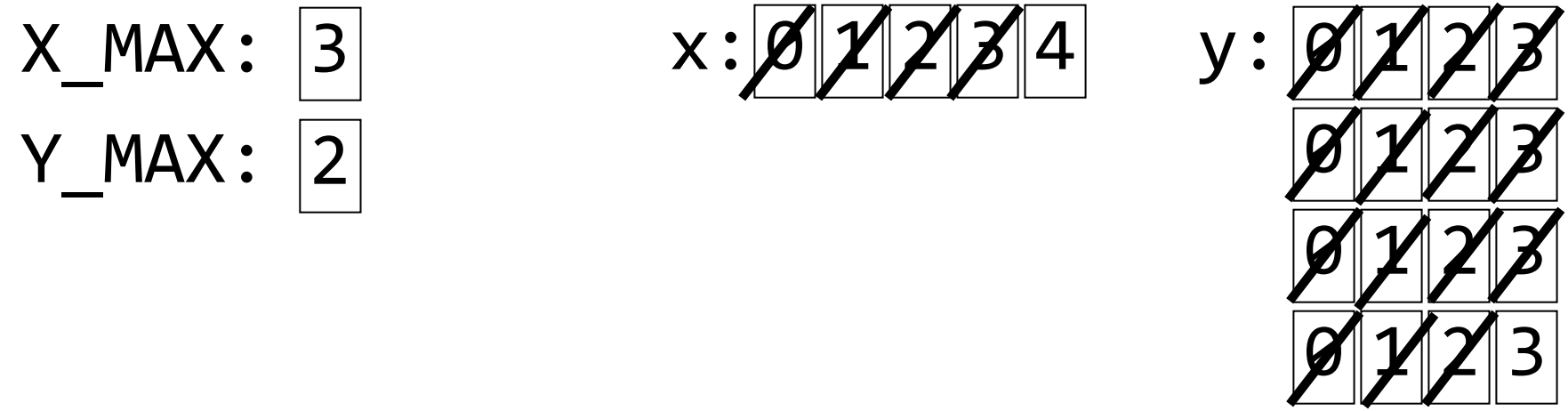
```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```



```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```



```
for (int x = 0; x <= X_MAX; x++)
{
    for (int y = 0; y <= Y_MAX; y++)
    {
        printf("(%d,%d)", x, y);
    }
}
```



gcc and C99

- gcc doesn't use all C99 features by default
- `gcc -std=c99`
- `gcc -std=c99 -Wall -o hello hello.c`

```
   ***
   ***
   ***
   ***
```

height: 4

width: 3

height rows

width "*" per row for all rows

```
for (int y = 0; y < height; y++)
{
  for (int x = 0; x < width; x++)
  {
      printf("*");
  }

  printf("\n");
}
```

```
*
**
***
****
```

height: 4

height rows

y "*" in row y

```
for (int y = 0; y < height; y++)
{
  for (int x = 0; x < y; x++)
  {
      printf("*");
  }

  printf("\n");
}
```

```
*
**
***
****
```

height: 4

height rows

y "*" in row y (1-based index)

```
for (int y = 1; y <= height; y++)
{
  for (int x = 0; x < y; x++)
  {
      printf("*");
  }

  printf("\n");
}
```