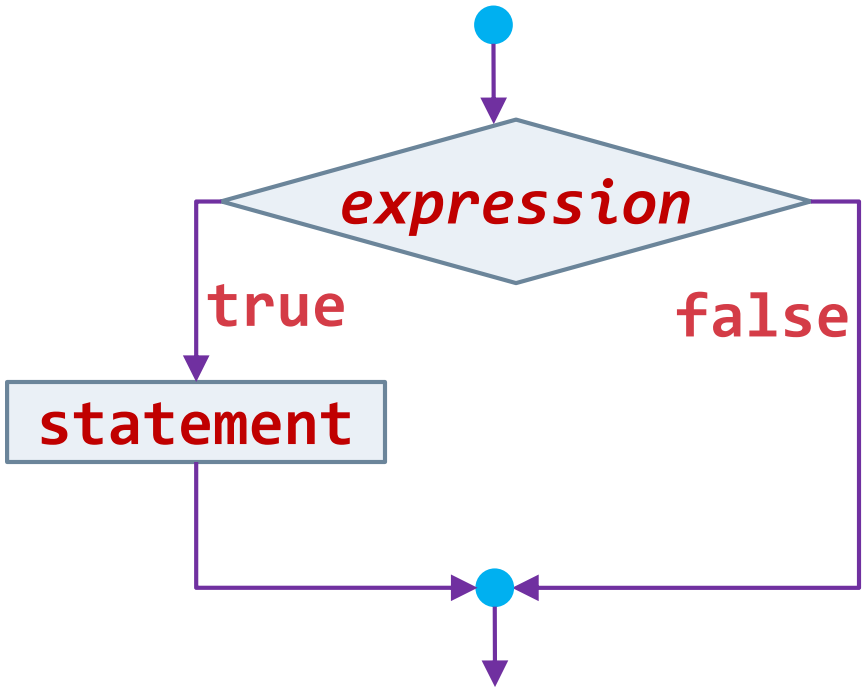
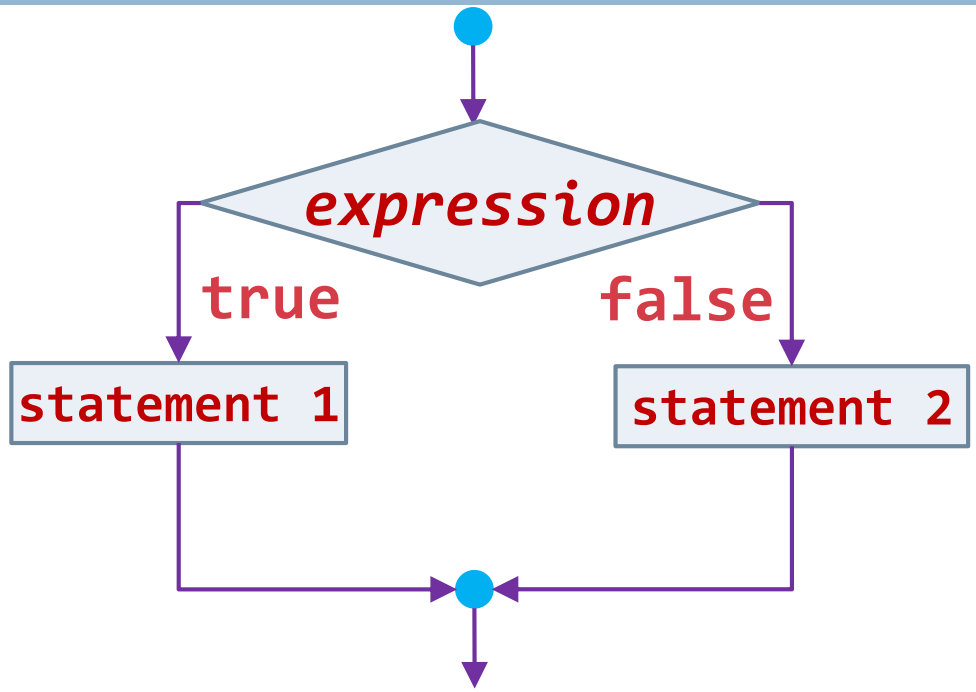
HIGH-LEVEL PROGRAMMING I

Selection: if and if-else

by Prasanna Ghali

Selection Structure






```
if (expression)
   statement
```

```
if (expression)
   statement 1
else
   statement 2
```

statement or
block of **statement**s delimited by braces

Meaning of C/C++ Statement

- **statement** is:
  - ;
  - ***expression*** followed by ;
  - Zero or more statements enclosed in opening brace { and closing brace }

```
d = sqrt(b*b-4.0*a*c);
```

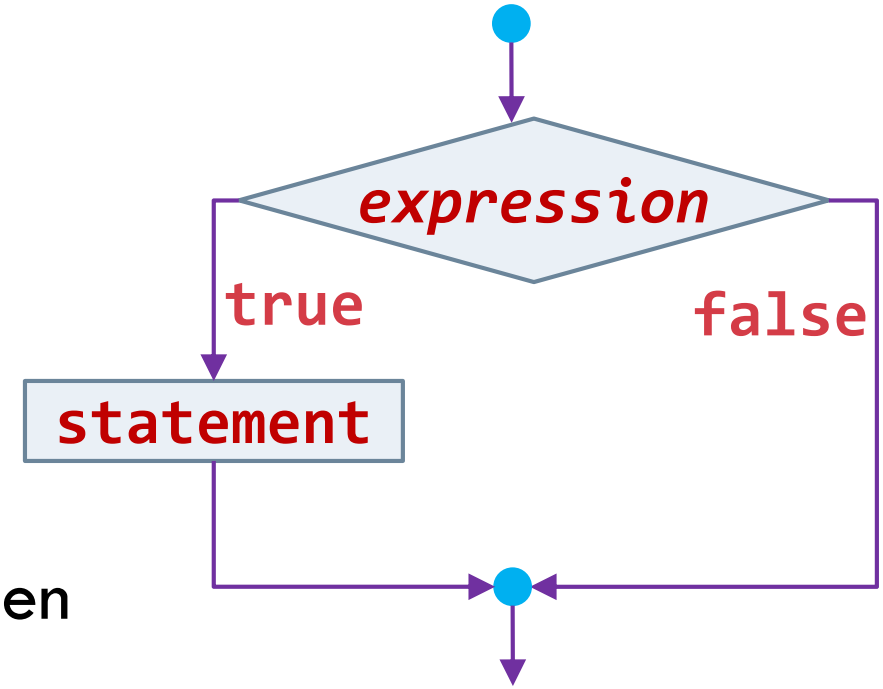
```
{
  // empty block
}
```

```
{
  w = x2 - x1;
  h = y2 - y1;
  d = sqrt(w*w + h*h);
}
```

One Way Selection: `if` Statement

- C/C++ Syntax:

```
if (expression)
   statement
```



- "If ***expression*** is `true`, then execute **`statement`**"
- "If ***expression*** is `false`, then *don't* execute **`statement`**"

if Statement: Example

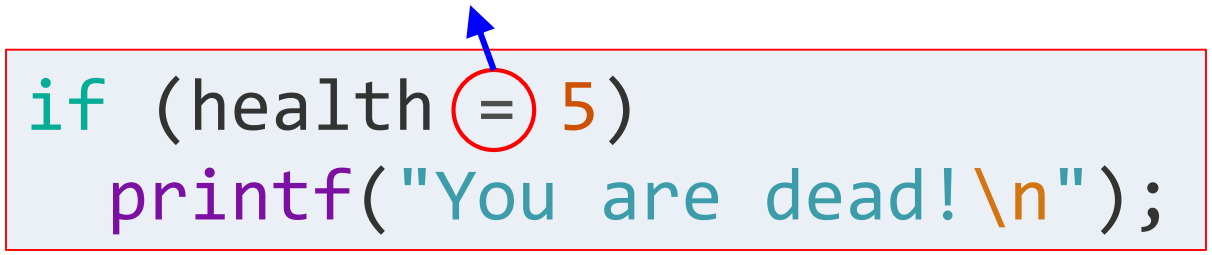
- Compute absolute value of a number

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

int abs_value = value;
if (abs_value < 0) {
  abs_value = -abs_value;
}

printf("Absolute value of %d is %d\n",
        value, abs_value);
```

`if` Statement: Common Error

- Common error:

Always `true`!!! Result of assignment expression is 5



```
if (health = 5)
  printf("You are dead!\n");
```

- More saner:

```
if (5 == health)
  printf("You are dead!\n");
```

`if` Statement: Syntax (1/4)

- Recall `if` statement syntax

  ```
  if (expression)
     statement
  ```

- **`statement`** syntax:
  - Zero expression followed by `;`
  - Expression followed by `;`
  - Zero or more statements enclosed { and closing brace }

```
if (a > b)
  ;
```

```
if (a > b)
  c = 10;
```

```
if (a > b) {
 int c = 10;
 a = b*c;
}
```

if Statement: Syntax (2/4)

- Block with single statement

```
if (a > b)
  printf("a = %d, b = %d\n", a, b);
```

- Block with multiple statements

```
if (a > b) {
  printf("a = %d, ", a);
  printf("b = %d\n", b);
}
```

`if` Statement: Syntax (3/4)

- Empty statements are valid!!!

```
if (a > b)
   ;
```

```
if (a > b) {

}
```

- Good habit to put single statement in braces!!!

```
if (a > b) {
  printf("a = %d, b = %d\n", a, b);
}
```

`if` Statement: Syntax (4/4)

- Another common error:

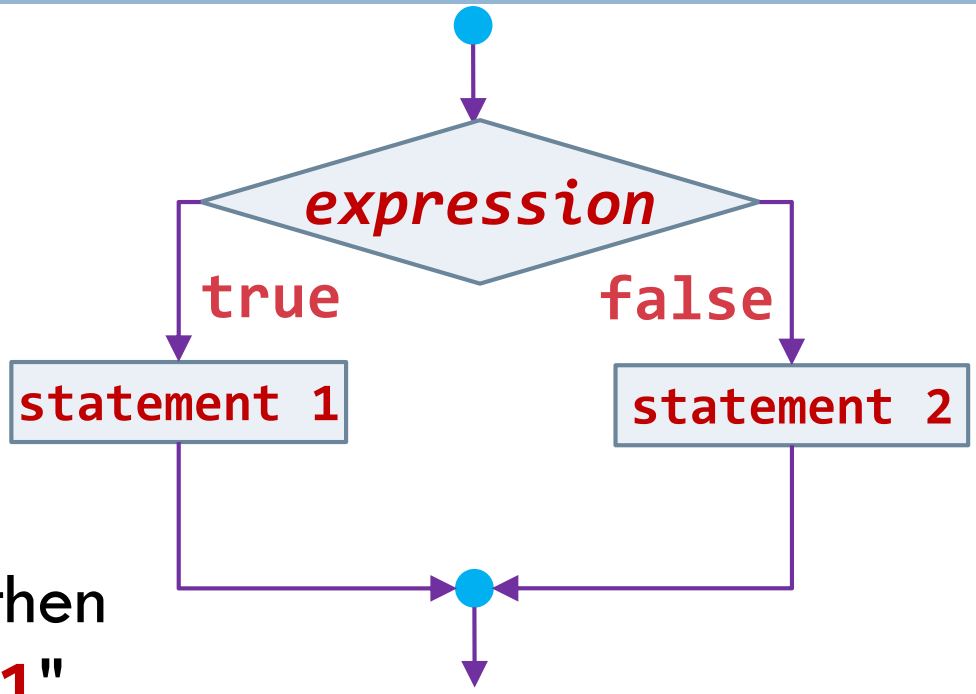
```
if (health == 0);
  printf("You are dead!\n");
```

Being *aware* when programming and carefully reading diagnostic messages from compiler will prevent these gotchas from spoiling your day

Two-way Selection Structure

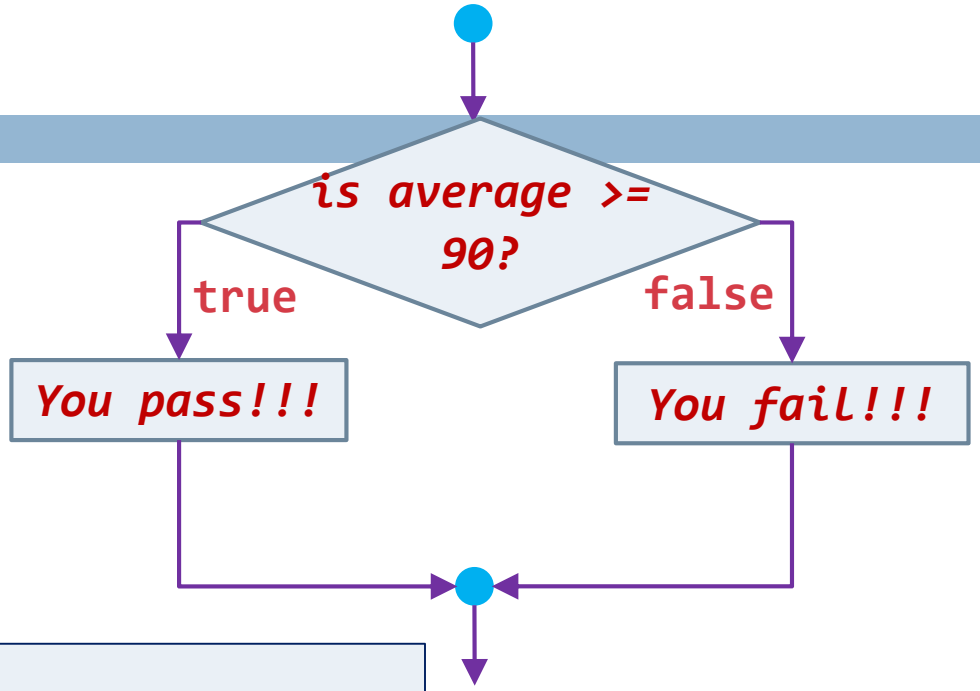
- C/C++ Syntax:

```
if (expression)
   statement 1
else
   statement 2
```



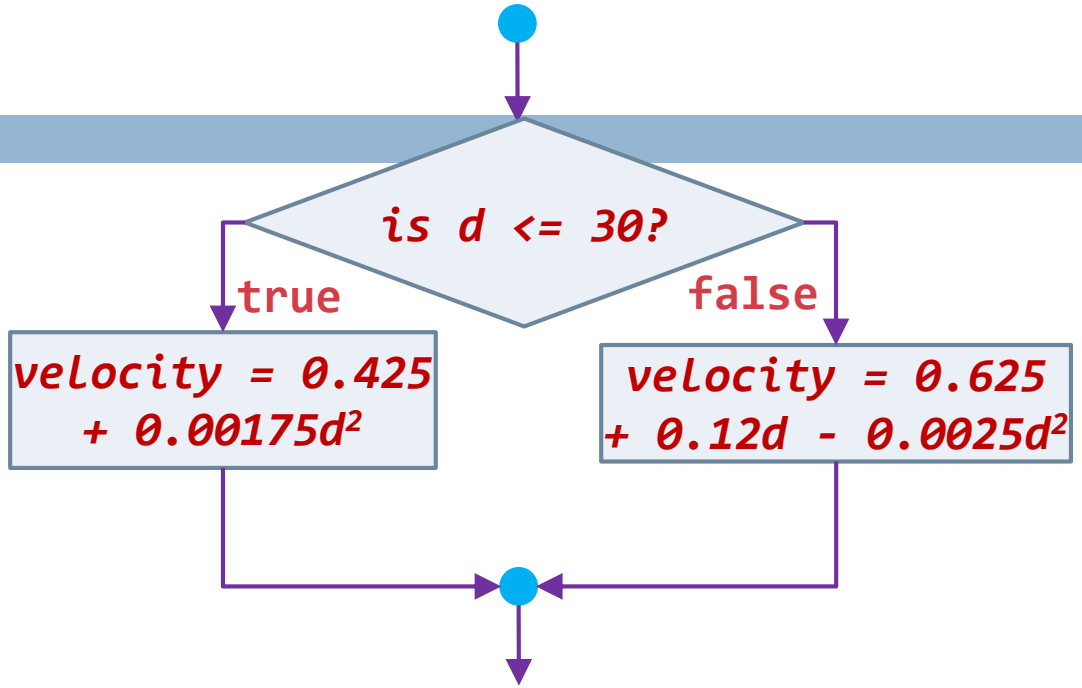
- "If ***expression*** is `true`, then *only* execute **`statement 1`**"
- "If ***expression*** is `false`, then *only* execute **`statement 2`**"

Two-way Selection Structure: Example One




```
int average;
printf("What is your score for HLP1?: ");
scanf(" %d", &average);

if (average >= 90) {
  printf("You passed, well done!\n");
} else {
  printf("See you next semester...\n");
}
```

Two-way Selection Structure: Example Two




```
...

if (d <= 30.0) {
  velocity = 0.425 + 0.00175*d*d;
} else {
  velocity = 0.625 + 0.12*d - 0.0025*d*d;
}
```

Multiple Selections: Nested `if` (1/7)

- Some problems require implementation of more than two alternatives
- Calculating interest on your bank balance

Checking account balance	Interest Rate
< $1,000	0%
$1,000 to $24,999.99	3%
$25,000 to $49,999.99	5%
>= $50,000	7%

Multiple Selections: Nested `if` (2/7)


```
double balance, int_rate;

if (balance < 1000.0) {
  int_rate = 0.0;
} else {
  if (balance < 25000.0) {
    int_rate = 0.03;
  } else {
    if (balance < 50000.0) {
      int_rate = 0.05;
    } else {
      int_rate = 0.07;
    }
  }
}
```

Checking account balance	Interest Rate
< $1,000	0%
$1,000 to $24,999.99	3%
$25,000 to $49,999.99	5%
>= $50,000	7%

Multiple Selections: Nested `if` (3/7)


```
double balance, int_rate;

if (balance >= 50000.0) {
  int_rate = 0.07;
} else {
  if (balance >= 25000.0) {
    int_rate = 0.05;
  } else {
    if (balance >= 1000.0) {
      int_rate = 0.03;
    } else {
      int_rate = 0.0;
    }
  }
}
```

Checking account balance	Interest Rate
< $1,000	0%
$1,000 to $24,999.99	3%
$25,000 to $49,999.99	5%
>= $50,000	7%

Multiple Selections: Nested if (4/7)


```
double balance, int_rate;

if (balance < 1000.0) {
  int_rate = 0.0;
} else {
  if (balance < 25000.0) {
    int_rate = 0.03;
  } else {
    if (balance < 50000.0) {
      int_rate = 0.05;
    } else {
      int_rate = 0.07;
    }
  }
}
```

```
double balance, int_rate;

if (balance >= 50000.0) {
  int_rate = 0.07;
} else {
  if (balance >= 25000.0) {
    int_rate = 0.05;
  } else {
    if (balance >= 1000.0) {
      int_rate = 0.03;
    } else {
      int_rate = 0.0;
    }
  }
}
```

Multiple Selections: Nested `if` (5/7)

- Alternative indentation that "saves space"

```
double balance, int_rate;

if (balance < 1000.0) {
  int_rate = 0.0;
} else {
  if (balance < 25000.0) {
    int_rate = 0.03;
  } else {
    if (balance < 50000.0) {
      int_rate = 0.05;
    } else {
      int_rate = 0.07;
    }
  }
}
```

```
double balance, int_rate;

if (balance < 1000.0) {
  int_rate = 0.0;
} else if (balance < 24999.99) {
  int_rate = 0.03;
} else if (balance < 49999.99) {
  int_rate = 0.05;
} else {
  int_rate = 0.07;
}
```

Multiple Selections: Nested if (6/7)


```
double balance, int_rate;

if (balance >= 50000.0) {
  int_rate = 0.07;
} else {
  if (balance >= 25000.0) {
    int_rate = 0.05;
  } else {
    if (balance >= 1000.0) {
      int_rate = 0.03;
    } else {
      int_rate = 0.0;
    }
  }
}
```

```
double balance, int_rate;

if (balance >= 50000.0) {
  int_rate = 0.07;
} else if (balance >= 25000.0) {
  int_rate = 0.05;
} else if (balance >= 1000.0) {
  int_rate = 0.05;
} else {
  int_rate = 0.07;
}
```

Multiple Selections: Nested `if` (7/7)

- Grading algorithm for an unknown course in some unknown school

```
if (average >= 90.0) {
  grade = 'A';
} else if (average >= 80.0) {
  grade = 'B';
} else if (average >= 70.0) {
  grade = 'C';
} else if (average >= 60.0) {
  grade = 'D';
} else {
  grade = 'E';
}
```

Grade point average	Letter grade
>= 90	A
80 to 89.99	B
70 to 79.99	C
60 to 69.99	D
< 60	E

Pairing an `else` with an `if` (1/3)

- How do you (or the compiler) know which `else` is paired with which `if`?
  - Remember there's no standalone `else` - every `else` must be paired with corresponding `if`
  - C/C++ are free form languages, so compilers ignore indentation - they see just sequence of tokens

```
if (average >= 90.f)
  grade = 'A';
else if (average >= 80.f)
  grade = 'B';
else if (grade >= 70.f)
  grade = 'C';
else
  grade = 'D';
}
```

```
if (attack > 20)
  if (damage <= 0)
    printf("No damage.\n");
else
  printf("You missed!\n");
```

Pairing an `else` with an `if` (2/3)

- In nested `if` statement, C associates an `else` with most recent `if` not already paired with `else`

```
if (attack > 20)
  if (damage <= 0)
    printf("No damage.\n");
else
  printf("You missed!\n");
```

```
if (attack > 20)
  if (damage <= 0)
    printf("No damage.\n");
  else
    printf("You missed!\n");
```

- Left code fragment is equivalent to right code fragment
- Known as *dangling-else* problem

Pairing an `else` with an `if` (3/3)

- What if you didn't mean this?
- Be explicit!!! Wrap `if` statement with curly braces!!!
  - Everything in braces becomes one (compound) statement, so inner `if` is "hidden" from `else`

```
if (attack > 20)
  if (damage <= 0)
    printf("No damage.\n");
  else
    printf("You missed!\n");
```

```
if (attack > 20) {
  if (damage <= 0)
    printf("No damage.\n");
} else {
  printf("You missed!\n");
}
```

Summary

- Syntactic meaning of C/C++ statement
- `if` statement and its applications
- `if else` statement and its applications
- Gotchas to watch out for when writing these statements
- Big gotcha is *dangle* `else` problem