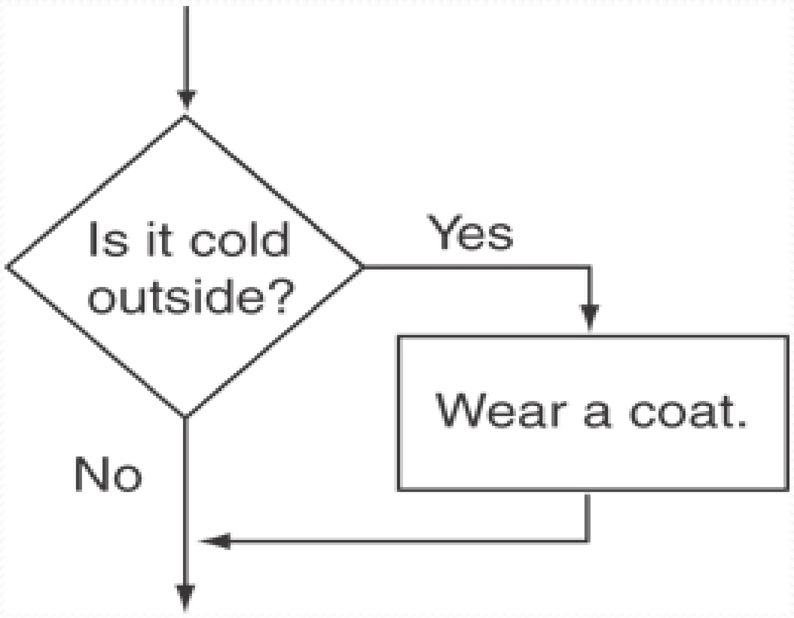
PFI lecture Flow of control statements

- “if” statement
  - Allows us to execute certain statements if the condition is met.
    - "If it is raining, take an umbrella."
    - "If the value of x is not zero, divide y by x"
- Syntax of if statement

```
if (expression)
        a single statement;

if (expression){
        one or more statements;
}
```

PFI lecture Flow Chart of if statement

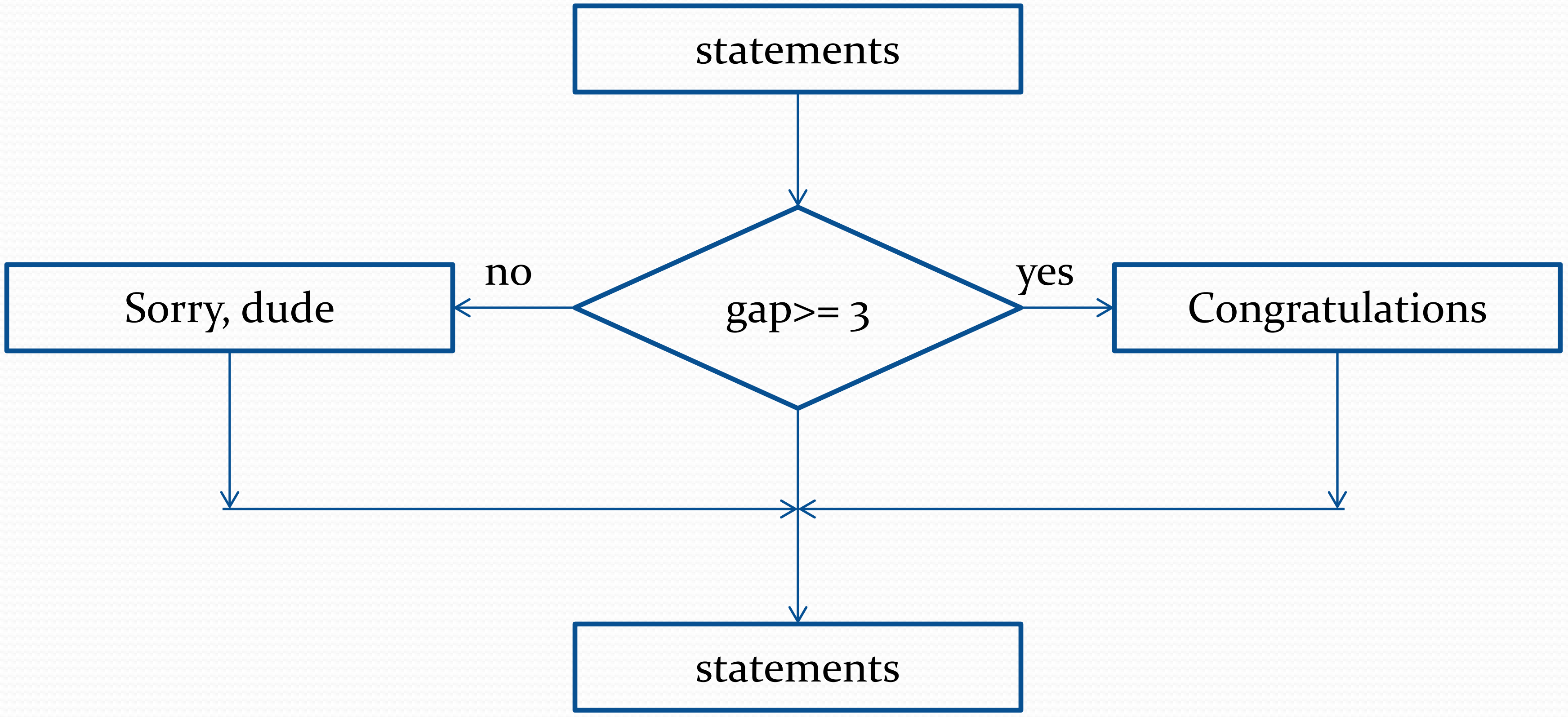


PFI lecture Flow of control statements

- “if else” statement
  - Allows us to execute certain statements if the condition is met and other statements if the condition is not met exclusively.
    - "If the GPA is 3.0 or more, say Congratulations; otherwise say Sorry dude"
- Syntax of if statement

```
if (expression)
          a single statement;
else
          a single statement;
if (expression){
          one or more statements;
}else{
          one or more statements;
}
```

PFI lecture Flow Chart of if else statement



Examples and Combinations

- Assigning letter grade example (grade is of type char and score is of type double:

```
if (score >= 90)
   grade = 'A';
else
   if (score >= 80)
      grade = 'B';
   else
      if (score >= 70)
         grade = 'C';
      else
         if (score >= 60);
            grade = 'D';
         else
            grade = 'F';
```

Examples and Combinations

- Assigning letter grade example (grade is of type char and score is of type double:

```
if (score >= 90)
   grade = 'A';
else if (score >= 80)
   grade = 'B';
else if (score >= 70)
   grade = 'C';
else if (score >= 60);
   grade = 'D';
else
   grade = 'F';
```

Examples and Combinations

- Assigning letter grade example (grade is of type char and score is of type double:

```
if (score >= 90)
   grade = 'A';
if (score < 90 && score >= 80)
   grade = 'B';
if (score < 80 && score >= 70)
   grade = 'C';
if (score < 70 && score >= 60)
   grade = 'D';
If (score < 60)
   grade = 'F';
```

Examples and Combinations

- Assigning letter grade example (grade is of type char and score is of type double:

```
if (score < 60)
   grade = ‘F';
else if (score < 70)
   grade = ‘D';
else if (score < 80)
   grade = 'C';
else if (score < 90);
   grade = ‘B';
else
   grade = ‘A';
```

Which one to use?

- We know the four example code fragments of assigning letter grade are logically the same, that is when executed the behavior is the same.
- Be the second one and the last one are perhaps easier for us to read and understand. Do you agree?

Discussions

- Note that x = 0 is an expression, a variable and a literal connected by = operator. So the following is ok syntactically

```
 if ( !(x = 0) ) // if x is not zero
     y/x;
```

- But it is not the same as

```
 if ( !(x == 0) )// if x is not zero
     y/x;
```

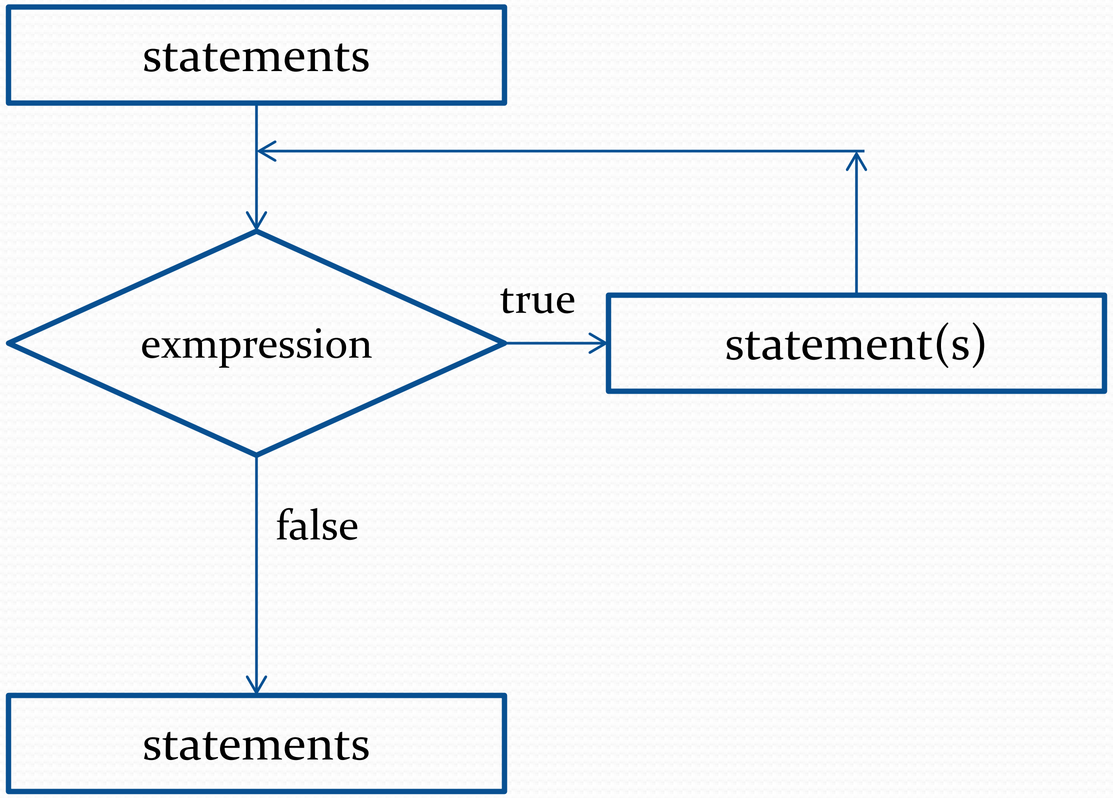
- DO NOT make such an error of using = for == in if statement!

PFI lecture Flow of control statements

- “while” loop statement
  - Allows us to repeat the execution of certain statements as long as the the condition is met.
    - “As long as you can still see, please keep digging deeper“
    - “As long as the value the variable is not 20, adding 1 to it"
- Syntax of while statement

```
while (expression)
        a single statement;
while (expression){
        one or more statements;
}
```

PFI lecture Flow Chart of while loop statement



Examples and Combinations

- Find the sum of 1+2+3+…+20:

```
int sum = 0;
int term = 1;
while ( term <= 20 ){
   sum = sum + term;
   term = term + 1;
}
cout << "sum = " << sum << endl;
```

Examples and Combinations

- Find the sum of the first 20 terms in the following sequence: 1,1,2,3,5,8,13,21,34,55…

```
int pre_term = 0;
int term = 1;
sum = 0;
count = 0;
while ( count < 20 ){
  sum = sum + term;
  count = count + 1;
  term = pre_term + term;
  pre_term = term - pre_term;
}
cout << "sum = " << sum << endl;int sum = 1;
```

Examples and Combinations

- Find the sum of the first 20 terms in the following sequence: 1,1,2,3,5,8,13,21,34,55…

```
int pre_term = 0;
int term = 1;
sum = 0;
count = 0;
while ( count < 20 ){
  sum = sum + term;
  count = count + 1;
  term = pre_term + term;
  pre_term = term - pre_term;
}
cout << "sum = " << sum << endl;int sum = 1;
```

Examples and Combinations

- Decide if an integer is a prime number. A Prime Number can be divided evenly only by 1, or itself. And it must be a whole number greater than 1.

```
int number;
int count;
bool found = false;
cin >> number;
if ( number> 1) {
   count = 2;
   while ( !found && count*count <= number ){
      if ( sum % count == 0 )
            found = true;
      else
            count = count + 1;
   }
   if (found)
            cout << number << " is not a prime number.\n ";
   else
            cout << number << " is a prime number." << endl;
} else
   cout << "Your input is " << number << " which is not greater that 1.\n“;
```

Examples and Combinations

- Listing all primes under 1000. Note: two loops nested!

```
int number = 2;
int count;
while (number <= 1000) {
   found = false;
   count = 2;
   while ( !found && count*count <= number ){
           if ( number % count == 0 )
                   found = true;
           else
                   count = count + 1;
   }
   if (!found){
           cout << number << " ";
   }

   number = number + 1;
}
```