

# Introduction to Programming

*with Java, for Beginners*

Do-while  
Continue  
Break

## Another loop: Do-While

```
do {  
    statement(s)  
} while (condition);
```

- Do the statement/block at least once
- Evaluate the *condition*. If it is
  - true: re-execute *statement(s)*; repeat step 2
  - false: we're done with the loop

```
int x = 0;  
do {  
    x = x + 1;  
}while (x < 3);
```

ESE112

1

## Break and Continue Statements

- *break* and *continue* are Java statements
- Are also “flow control” statements
  - if, while, do-while, for, return
- A break “breaks you out” of the closest enclosing loop
- A continue is a shortcut to the next iteration of the loop
- A loop may have
  - Zero or more break statements
  - Zero or more continue statements

ESE112

2

## while-loop with break, continue

```
while (condition1){  
    . . .  
    if (condition2)  
        continue; // go up and re-evaluate condition1  
    if (condition3)  
        break; // exit the loop  
    . . .  
    . . .  
}  
// after a break statement, execution resumes here
```

ESE112

3

### while-loop with break, continue example

```
int x = 1;
while (x <= 10){
  if (x % 2 == 0){
    System.out.println(x);
    break;
  }
  x = x + 1;
}

int x = 1;
while (x <= 10){
  if (x % 2 == 0){
    System.out.println(x);
    continue;
  }
  x = x + 1;
}
```

What will happens with break vs. continue ?

ESE112

4

### for-loop with break, continue

```
for (expr1; condition1; expr2){
  . . .
  if (condition2)
    continue; // evaluate expr2, then condition1
  if (condition3)
    break; // exit the loop
  . . .
}
// after a break statement, execution resumes here
```

ESE112

5