

Apple Campus
One Infinite Loop



Unit 8

- The `while` Loop
- Extending the `Vic` class
- Examples



Loops in Java

A structure that allows the programmer to repeat something while a condition is true. (also called “iteration”)

Three kinds of loops in Java

1. The “while” loop // we will see today
2. The “for” loop // we will see next lesson
3. The “do-while” loop // not on AP exam...in your book



Loops in Java

1. The “while” loop – used when we want to repeat something **while** a condition remains true.

Syntax:

```
while (condition)
{
    statement ;
    statement ;
    ...
}
```

The program will execute these statements while the condition remains true.

When the condition becomes false, the program will resume after the curly brace.



Example

Rewrite Program 2 from Unit 7 using less than 10 lines of program code.



```
public class Program2
{
    public static void main (String[ ] args)
    {
        Vic one = new Vic();
        Vic two = new Vic();

        while(one.seesSlot() && two.seesSlot())
        {
            if (!two.seesCD() && one.seesCD())
            {
                one.takeCD();
                two.putCD();
            }

            one.moveOn();
            two.moveOn();
        }
        ... // Repetition done in the loop!!
    }
}
```



Example: Explain what this application program does.

```
public class Mystery
{
    public static void main (String [] args)
    {
        Vic.reset(args);
        Vic v = new Vic();

        while(Vic.stackHasCD())
        {
            v.putCD();
            v = new Vic();
        }
    }
}
```



The while Loop (cont'd)

- Example:

```
// Returns the sum of the first
// n integers
public int sumAll (int n)
{
    int p = 1;
    int sum = 0;
    while ( p < n )
    {
        sum += p;
        p++;
    }
    return sum;
}
```

Initialization

Testing

Change



The while Loop (cont'd)

- Initialization: The variables tested in the condition must be initialized to some values. If the condition is false at the outset, the loop is never entered.
- Testing: The condition is tested before each iteration. If false, the program continues with the first statement after the loop.
- Change: At least one of the variables tested in the condition must change within the body of the loop.



Extending the Vic

Shortcoming: Vic does not have a method to go back to the front of the sequence (the first slot).

Extension: called **inheritance** in OOP.

Allows us to keep all of the qualities of a class, plus add any new ones.

Introducing the DualDirectionVic

Allows us to keep all of the qualities of a `Vic`, plus add any new ones.

The new ones: 1. A method to back up to the first slot.
2. A method to go to the last slot.
3. A boolean method to check for first slot.



```
public class DualDirectionVic extends Vic
{
    private String itsFirstPosition;

    public DualDirectionVic()
    {
        super();
        itsFirstPosition = getPosition();
    }

    public void goToFirst()
    {
        while(!seesFirstSlot())
            backUp();
    }

    public void goToLast()
    {
        while(seesSlot())
            moveOn();
        backUp();
    }

    public boolean seesFirstSlot()
    {
        if(itsFirstPosition.equals(getPosition()))
            return true;
        else
            return false;
    }
}
```