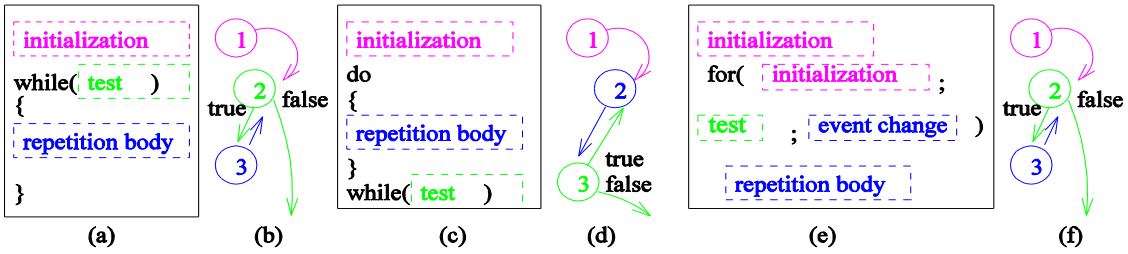
Solution to Practice Problems: Loops



---

1. Understanding code

Draw a representation of what the computer's memory looks like at the end of each of these programs, and provide the display result on screen (by System.out.print or System.out.prinln).

```
public class Simple-While {
  public static void main(String [] args) {
    int x = 1;
    while(x < 10) {
      System.out.println("x = " + x);
      x++;
    }
  }
}
```

x: starts at 1, changes to 2, then 3, then 4, then 5, …, then 9, then 10

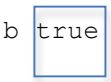
answer

x: 10

screen

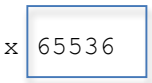
```
x = 1
x = 2
x = 3
x = 4
x = 5
…
x = 8
x = 9
(NOTE:  x = 10 does NOT get printed by this program)
```

```
public class Infinite-While {
public static void main(String [] args) {
      boolean b = true;
      while(b) {
      System.out.println("are we there yet?");
      }
   }
}
```



screen
are you there yet?
…
(infinite)

```
public class Complex-Update {
public static void main(String [] args) {
      int x = 2;
      while(x < 1000) {
         System.out.println("x = " + x);
         x = x * x;
      }
   }
}
```



screen
x = 2
x = 4
x = 16
x = 256

```
public class Infinite-For {
public static void main(String [] args) {
      for( ; true; ) {
         System.out.println("are we there yet?");
      }
   }
}
```

screen
are you there yet?
…
(infinite)

```
public class Complex-Update-For {
public static void main(String [] args) {
      for(int i = 2; i < 1000; i = i * i) {
            System.out.println("i = " + i);
      }
   }
}
```

i 65536

screen
i = 2
i = 4
i = 16
i = 256

```
public class Factor {
  public static void main(String [] args) {
        int number = 13;
        int factor = 2;
        while (number % factor != 0) {
               factor++;
        }
        System.out.println("First factor: " + factor);
  }
}
```

number: 2
factor: starts at 2, then 3, then 4, …, then 12, then 13
screen: First factor: 13

```
public class Sum {
  public static void main(String [] args) {
    int sum = 0, n =12345;
    while (n > 0) {
        sum += n % 10;   // add last digit to sum
        n = n / 10;      // remove last digit
    }
    System.out.println("sum = " + sum);
  }
}
```

n: starts at 12345, then 1234, then 123, then 12, then 1, then 0
sum: starts at 5, then 9, then 12, then 14, then 15
screen: sum = 15