

CSE 142 Final Cheat Sheet

```

for (initialization; test; update) {
    statement(s);
    ...
}

if (test) {
    statement(s);
} else if (test) {
    statement(s);
} else {
    statement(s);
}

```

```

while (condition) {
    statement(s);
}

public static type name(parameters) {
    statement(s);
    ...
    return expression;
}

```

| Math Method | Description |
|-----------------------------------|----------------------------|
| Math.abs(<i>value</i>) | absolute value |
| Math.min(<i>v1</i> , <i>v2</i>) | smaller of two values |
| Math.max(<i>v1</i> , <i>v2</i>) | larger of two values |
| Math.round(<i>value</i>) | nearest whole number |
| Math.sqrt(<i>value</i>) | square root |
| Math.pow(<i>b</i> , <i>e</i>) | base to the exponent power |

| Random Method | Description |
|-----------------------|--|
| nextInt(<i>max</i>) | random integer from 0 to <i>max</i> -1 |

| String Method | Description |
|--|---|
| contains(str) | true if this string contains the other's characters inside it |
| endsWith(str), startsWith(str) | true if this string starts/ends with the other's characters |
| equals(str) | true if this string is the same as <i>str</i> |
| equalsIgnoreCase(str) | true if this string is the same as <i>str</i> , ignoring capitalization |
| indexOf(str) | index in this string where given string begins (-1 if not found) |
| length() | number of characters in this string |
| replace(str1 , str2) | replace all occurrences in this string of <i>str1</i> with <i>str2</i> |
| substring(i , j) | characters in this string from index <i>i</i> (inclusive) to <i>j</i> (exclusive) |
| toLowerCase(), toUpperCase() | a new string with all lowercase or uppercase letters |
| charAt(i) | returns char at index <i>i</i> |

| Scanner Method | Description |
|-----------------|--|
| nextInt() | reads/returns input token as <code>int</code> |
| next() | reads/returns input token as <code>String</code> |
| nextDouble() | reads/returns input token as <code>double</code> |
| nextLine() | reads/returns line as <code>String</code> |
| hasNextInt() | returns <code>true</code> if there is a next token and it can be read as an <code>int</code> |
| hasNext() | returns <code>true</code> if there is a next token to read |
| hasNextDouble() | returns <code>true</code> if there is a next token and it can be read as a <code>double</code> |
| hasNextLine() | returns <code>true</code> if there is a next line to read |

Declaring and using Arrays

```
type[] name = new type[length];  
name[index] = value;
```

Classes

Field *(data inside each object)*

```
private type name;
```

Method *(behavior inside each object)*

```
public type name(parameters) {  
    statements;  
}
```

Constructor *(code to initialize new objects)*

```
public className(parameters) {  
    statements;  
}
```

toString method *(called when an object is printed)*

```
public String toString() {  
    code that produces/returns a String;  
}
```

Inheritance

```
public class name extends superclass {  
}
```

Critter classes

```
public class name extends Critter {  
    fields  
  
    constructor  
  
    public boolean eat() {  
        statement(s) that return true (eat) or false (don't eat);  
    }  
  
    public Attack fight(String opponent) {  
        returns either Attack.ROAR, Attack.POUNCE, or Attack.SCRATCH;  
    }  
  
    public Color getColor() {  
        statement(s) that return a Color;  
    }  
  
    public Direction getMove() {  
        statement(s) that return either Direction.NORTH, Direction.SOUTH,  
        Direction.EAST, Direction.WEST, or Direction.CENTER;  
    }  
  
    public String toString() {  
        statement(s) that return a String;  
    }  
}
```