Item creations

## Class Item

Should have

* **Properties**:
  + **id** - generated when the item is created
  + **name** - provided in constructor - must be a string between 3 and 50 letters
  + Throw error if [name] does not meet requirements
* **Methods**:
  + **getItemInfo()**
    - method returning a string in the following format - ‘Item [id] - [name]’

## Class Weapon

**Extends Item** and adds the following

* **Properties**:
  + **attack** - provided in constructor - must be a number between 1 and 30000
  + **damageType** - provided in constructor - must be one of the following strings - physical, poison, fire, water, air, earth
  + **twoHanded** - provided in constructor - must be a boolean
  + **chance** - not provided from constructor - must be a random number between 5 and 50
  + Throw error if [attack], [damageType] or [twoHanded] does not meet requirements
* **Methods**:
  + **getItemInfo()**
    - reuse from **Item** and add - ' has [attack] of [damageType] damage' to the string

### Class Sword

**Extends Weapon** and adds the following

* **Properties**:
  + **cripple** - not provided from constructor - based on [twoHanded] state. Set the value of the [twoHanded] property
  + **bleed** - not provided from constructor- Set the value of the [twoHanded] property
* **Methods**:
  + **getItemInfo()**
    - reuse from **Weapon** and add - ' and has [chance]% to [if cripple use the string “cripple”, if bleed use the string “bleed”]'

### Class Bow

**Extends Weapon** and adds the following

* **Properties**:
  + Bows can only be [twoHanded]
  + **arrowType** - provided in constructor - must be one of the following strings - normal, special
  + **pierce** - not provided from constructor - must be boolean. If [arrowType] is normal, set to true, if not, set to false
  + **critical** - not provided from constructor - must be boolean. If [arrowType] is special, set to true, if not, set to false
  + Throw errors if [twoHanded] is false or if [arrowType] does not meet requirements
* **Methods**:
  + **getItemInfo()**
    - reuse from **Weapon** and add - ' and has [chance]% to apply [if pierce use the string “pierce”, if critical use the string “critical”]'

### Class Staff

**Extends Weapon** and adds the following

* **Properties**:
  + Staves can only be [twoHanded] and cannot have physical [damageType]
  + **burn** - not provided from constructor - must be boolean. If [damageType] is fire, set to true, if not, set to false
  + **poison** - not provided from constructor - must be boolean. If [damageType] is poison, set to true, if not, set to false
  + **cold** - not provided from constructor - must be boolean. If [damageType] is water, set to true, if not, set to false
  + **electrify** - not provided from constructor - must be boolean. If [damageType] is air, set to true, if not, set to false
  + **tremor** - not provided from constructor - must be boolean. If [damageType] is earth, set to true, if not, set to false
  + Throw errors if [twoHanded] is false or if [damageType] does not meet requirements
* **Methods**:
  + **getItemInfo()**
    - reuse from **Weapon** and add - ' and has [chance]% to apply [if burn use the string “burn”, if poison use the string “poison”, if cold use the string “cold”, if electrify use the string “electrify”, if tremor use the string “tremor”]'

## Class Armor

**Extends Item** and adds the following

* **Properties**:
  + **defense** - provided in constructor - must be a number between 1 and 50000
  + **resistance** - provided in constructor - must be one of the following strings - physical, poison, fire, water, air, earth
  + **chance** -not provided from constructor - must be a random number between 10 and 100
  + Throw error if [defence] or [resistance] does not meet requirements
* **Methods**:
  + **getItemInfo()**
    - reuse from **Item** and add - ' has [defense] defense and [chance]% [resistance] resistance' to the string

### Class Helm

**Extends Armor** and adds the following

* **Properties**:
  + **attractiveness** - provided in constructor - must be within [-5; 5]
  + Throw error if [attractiveness] does not meet requirements
* **Methods**:
  + **getItemInfo()**
    - reuse from **Armor** and add - ' and adds [attractiveness] attractiveness' to the string

### Class Boots

**Extends Armor** and adds the following

* **Properties**:
  + **speed** - provided in constructor - must be a number between 1 and 10
  + Throw error if [speed] does not meet requirements
* **Methods**:
  + **getItemInfo()**
    - reuse from **Armor** and add - ' and adds [speed] speed' to the string

### Class Gloves

**Extends Armor** and adds the following

* **Properties**:
  + **crafting** - provided in constructor - must be a number between 1 and 10
  + Throw error if [crafting] does not meet requirements
* **Methods**:
  + **getItemInfo()**
    - reuse from **Armor** and add - ' and adds [crafting] crafting' to the string

### Class Robe

**Extends Armor** and adds the following

* **Properties**:
  + **reputation** - provided in constructor - must be a number between 1 and 10
  + Throw error if [reputation] does not meet requirements
* **Methods**:
  + **getItemInfo()**
    - reuse from **Armor** and add - ' and adds [reputation] reputation' to the string

## Class Consumable

**Extends Item** and adds the following

* **Properties**:
  + **heals** - provided in constructor - must be a boolean
  + **type** - provided in constructor - must be one of the following strings - minor, medium, big
  + **effect** - not provided from constructor - based on type:
    - minor - random number between 1 and 10
    - medium - random number between 11 and 20
    - big - random number between 21 and 30
  + Throw error if [type] does not meet requirements
* **Methods**:
  + **getItemInfo()**
    - reuse from **Item** and add - based on the [heals] boolean the string can be ' it is [type] potion and heals for [effect]' if true or ' it is [type] potion and damages for [effect]'