# Java OOP Basics Exam

# –

# The Younger Scrolls EIGHT

## Overview

For the last 3 years, gamers have been pressuring Dethesba Handworks company for a new “The Younger Scrolls” game and finally in 2018 they decided to announce, “The Younger Scrolls EIGHT” which will come in early 2019(hopefully). For their next game, Dethesba promised that their new game will be bigger, twice as immersive and with even more Easter eggs than its predecessor. Dethesba also made the big announcement that there will be no more bugs and glitches in their games because they have hired a “super-developer”, a developer with ability to implement the character creation system, the guilds and the battle system in a mere 6 hours.

Yes… They hired you!

## Task I: Structure

Your first task will be to **implement the structure of the heroes** that will roam Winterset isles (a continent in the younger scrolls).

### Interfaces

For their game Dethesba has given you two interfaces, Hero and Runnable.

#### Runnable

The “Runnable” interface will represent the whole game that Dethesba hired you to write.

Runnable has just one method and that is

**void run() –** This will be the entry point of the game. It will also hold all the game`s logic.

#### Hero

The Hero interface is a representation of all the heroes which will be in the game. Hero has the following methods:

**String getName() –** returns the name of the hero.

**int getHealth() –** returns the health of the hero.

**double getOffense()** – returns the offense of the hero.

**double getDefense()** – returns the defense of the hero.

**double getTotalPoints()** – returns the total points of the hero.

**boolean isDead()** - returns “true” if the hero is dead.

**void attack(Hero hero)** - attack another character.

**void receiveDamage(double amount)** – receive damage equal to the double amount.

### Characters

Every character in TYS eight has a **name(String), magicka(int), fatigue(int), health(int)** and **heroType(String).** An advice that Dethesba gave you though is **to retain the order they are given in,** so as not to create any aditional bugs (Remember their promise, no more bugs! ).

Every hero has Offense and Defense abilities which will differ depending on the hero`s birth sign (**information** **below**).

### Birth signs

A hero is created with a strength, endurance or willpower birth sign.

#### Strength

Strength heroes by nature are there to deal a lot of physical damage but they can deal magic damage with their weapons as well. Every strength hero has magicDamage(double) equal to 50% of the their magicka.

Heroes created under the strength birth sign are:

##### **Redguard**

Redguards hail from the great desert province of Lefremmah. They are descended from an extensive line of warriors and mystic seers. Legend has it that Redguards are innately more proficient with the use of weaponry than any other race.

Redguards have the hero type **“REDGUARD”.**

##### Khajiit

Khajiit are one of the beast races which inhabit the continent of Leirmat. They are cat-like creatures who can slay an entire army without anyone noticing.

Khajiit have the hero type **“KHAJIIT”.**

#### Endurance

Endurance heroes are tanks by nature. Therefore, they have magicResistance(double) equal to 40% of their maigcka. Upon creation endurance heroes also get their health DOUBLED.

Heroes created under the endurance birth sign are:

##### **Orc**

The Orsimer (more commonly known as Orcs), are the native people of the Nairaghtorw and Liatnogard Mountains. Orcs are known to be one of the greatest warriors and smiths in the province of Leirmat.

Orcs have the hero type **“ORSIMER”.**

##### Nord

Nords, also known as Sons of Snow are a race of brave men and women from the province of Miryks.

Nords have the hero type **“NORD”.**

#### Willpower

Willpower heroes fight with spells and since they have no use of fatigue they have found a way to turn it into **spellPentration (double) which is equal to 20% of the hero`s fatigue**;

Heroes created under the willpower birth sign are:

##### Breton

Bretons are a race of both human and elven ancestry. They populate the province of High Rock and are excellent mages.

Bretons have the hero type **“BRETON”.**

##### Dunmer

The Dunmer, more commonly referred to as Dark Elves, are the dark-skinned elves originating from the province of Dniworrom. They are known to be strong, intelligent, quick, aloof, and reserved, but some consider them as ill-fated.

Dunmer have the hero type of, yes you guessed it, **“DUNMER”.**

### Guild

Whoa, having so many heroes they need place to live right?! This is where a guild comes in place. The guild provides home for all these heroes. Each guild has a **collection of heroes** and a **name(String).**

### Province

The province is a continent where all guilds are situated. They are the place where all the fights happen, the place where all guilds are set up and all heroes are born.

A province has a **collection of guilds and a name(String)**

In the younger scrolls game there will be only one province and Dethesba gave you the honor of giving it its name!

For this task you **don’t have** to provide a correct result, only correct design. Be cautious of using if-else or switch statements. Utilize **inheritance**, **polymorphism** and **abstract classes/methods** to achieve good design.

### Constructors

Implement all **class constructors**, with the **parameters** in the **EXACT** **given order** and the **EXACT given types**. Zero tests will help you with this.

### Getters and Setters

Provide getters and/or setters where you think you might needed them, but do not break the principle of encapsulation or in other words don’t expose things you do not need to expose. The naming convention that you must follow is **camelCase.**

**Example: public void doSomething();**

### Judge

Don't forget to include a **Main class** with a **main method** in your project when submitting your solution in judge. You are allowed to create and use new methods and classes in your solution as long as they conform to good OOP practices.

## Task II: Business Logic

The business logic of the program should function in the way specified below.

### Hero Methods

Provide an implementation for the methods:

* **double getOffense()**
* **double getDefense()**

As mentioned above, each birth sign derives its offense and defense in a different way.

If the hero`s birth sign is **strength:**

* Offense is equal to: **125% of the hero`s fatigue + 30% of the health + the hero`s magic damage.**
* Defense is equal to: **the hero`s health + 10% of the hero`s fatigue.**

If the hero`s birth sign is **endurance:**

* Offense is equal to: **the fatigue of the hero.**
* Defense is equal to: **the health of the hero + 60% of the hero`s fatigue + the hero`s magic resistance.**
  + Orcs have **additional defense equal to 10% of the birth signs’ defense.**

If the hero`s birth sign is **willpower:**

* Offense is equal to: **180% of the hero`s magicka + the hero`s spell penetration.**
* Defense is equal to: **the health of the hero.**
* **boolean isDead() –** should **“true”** if the hero`s health is less than 1**.**
* **double getTotalPoints()** – return the sum of the hero`s offense and defense.
* **void attack(Hero hero)** –The attacked character should get **their health reduced by the floored amount of the attacker`s offense**.
* **void receiveDamage(double amount) –** reduce health **by the floored value of amount.**

### Guild Methods

Provide the following public methods:

* **String addHero(Hero hero) –** 
  + If a hero with the same name does **NOT exist**:
    - Add the hero to the collection and return “**Added hero: {heroName}**”
  + If a hero **with the same name exists**:
    - Check if the provided hero from the method addHero(Hero hero) is **stronger than the one already stored**.This is done **by comparing their total points.**
      * If the **provided hero`s total points are more than the total points of the stored hero** - Replace **the stored hero with the provided one** and

return “**Updated hero: {heroName}**”.

* + - * If the **provided hero`s total points are less than or equal to the total points of the stored hero –** return “**Hero {heroName} can not be replaced by a weaker one.**”
* **String removeHero(Hero hero) –** remove the hero with the given name from the guild and return “**Successfully removed hero [{heroName}] from guild {guildName}**”.
* **Hero getHeroByName(String heroName) –** retrieve the hero with the given name from the guild.
* **Long getGuildSize() –** retrieve the number of heroes in the guild.
* **double getGuildPower() –** return the sum of the total points of the heroes in the guild – the number should be formatted to two decimal places.
* **String getGuildSpecialization() –** returns “Willpower”, “Endurance” or “Strength” depending on the **birth sign under which the majority of the heroes in the guild are created**.
  + Example: If we have a guild with a Breton(willpower), Orc(endurance) and Dunmer(willpower) in our guild the method should return “Willpower” because the majority of the heroes in this guild are born under the “Willpower” birth sign.
    - If we have a guild who has birth signs with equal amounts of heroes created under them, you should return the birth sign whose sum of the total points of the heroes created under it is bigger.

### Province Methods

Provide the following public methods:

* **String addGuild(Guild guild) -** Add a guild to the collection of guilds and return “**Added Guild: {guildName}**”.
* **String removeGuild(Guild guild) –** remove the guild with the given name from the province and return “**Removed guild [{guildName}] with {guildSize} members.**”
* **Guild getGuildByName(String guildName) -** retrieve the guild with the given name from the province.
* **boolean contains(String guildName) –** return “true” if the province contains a guild with the given name.

## Task III: I / O (Input / Output)

### Commands

The younger scrolls game should support the following commands.

* **CREATE\_PROVINCE: {provinceName}**
  + Create a province with the given name and print – **“Created province {provinceName}”**
  + If a province with such name already exists print – **“Province with name {provinceName} already exists!”**

**#NOTE: After successfully creating a new province, the newly created province should be automatically selected!**

* **SELECT: {provinceName}**
  + Select a province and print **“Province {provinceName} selected!”**
  + If a province with such name does not exist print **“Province {provinceName} does not exist”**
  + If a province with the current name has already been selected print **“Province %s has already been selected!”**

#### Important

The Younger Scrolls game should apply all commands mentioned below **on the currently selected province!**

* If no province is selected before executing any of the commands below you should print **“No province selected!”**
* **ADD\_HERO: {guildName} {heroType} {heroName} {health} {fatigue} {magicka}**
  + If a guild with such name does not exist create it.
  + If the hero type does not exist.
    - Print “**No such hero type!**”
  + If health, fatigue or magicka are less than 1.
    - Print “**Invalid character stats!”**
  + If everything is okay, add the hero to the guild and print the result of adding the hero.
* **ADD\_GUILD: {guildName}**
  + If a guild with the given name already exists
    - Print - “**Guild already exists.**”
  + If a guild with the given name does not exist – add the guild to the province.
    - Print - “**Added Guild: {guildName}**”
* **DETAILS: {guildName}**
  + If the guild does not exist print “**Guild [{guildName}] does not exist.**”
  + If the guild exists, you should print:

**“Guild: {guildName}”**

**“###Heroes: ”**

**“Hero: {heroName}, Offense: {offense}, Defense: {defense}” -** for each hero in the guild. Offense and defense should be formatted to **two decimal places.**

* The heroes should be **sorted by their total points in descending order.** If two or more heroes have the same total points **sort them alphabetically by their name**.
  + If there are no heroes in the guild you should print “**###Heroes: None**”
* **DETAILS: {guildName} {heroName}**
  + If a guild with such name does not exist, print “**Guild [{guildName}] does not exist.**”
  + If a hero with such name does not exist in the guild, print “**No such hero in this guild!**”
  + If the guild exists and the hero does too, you should print the following:

**“Hero: {heroName}, Type: [{heroType}]”**

**“#Stats:”**

**“Health: {health of the hero}”**

**“Fatigue: {fatigue of the hero}”**

**“Magicka: {magicka of the hero}”**

* + If the hero is created under the “Endurance” birth sign you should also print his birth sign bonuses.

**“#Endurance bonuses:”**

**“Magic resistance: +{magicResistance} defense.” –** formatted **to two decimal places.**

* + The same applies for heroes under the “Strength” births sign.

**“#Strength bonuses:”**

**“Magic damage: +{magicDamage} offense.” -** formatted **to two decimal places.**

* **REMOVE: {guildName}**
  + Remove a guild with a given name from the province and print - **“Removed guild {guildName} with {number of heroes in guild} members.”**
  + If a guild with such name does not exist, print **“Guild {guildName} does not exist.”**
* **REMOVE: {guildName} {heroName}**
  + Remove a hero with the given name from the guild with the given name and print **“Successfully removed hero [{heroName}] from guild {guildName}”**
  + If a guild with such name does not exist, print **“Guild {guildName} does not exist.”**
  + If a hero with such name does not exist, print “**No such hero in this guild!**”
* **FIGHT: {guild1Name} {hero1Name} {guild2Name} {hero2Name}**
  + Fight is the command where two heroes will fight until one of them is dead.
  + When the two heroes start fighting **the hero with less health** should be attacking the other hero.
    - Example: **if Hero A has less health than Hero B -> Hero A attacks Hero B**.
  + If two **heroes have the same health** the hero **with less offense** should attack the other one.
    - Example: If Hero A and Hero B have the same health but **Hero A has less offense**, **Hero A should attack Hero B.**
  + The fighting ends when one of the heroes has **less than 0 health.**
  + After the battle you should print - **“Hero {winner hero name} sliced hero {killed hero name}.”** and the hero who **got killed should be removed from the guild.**
  + If the two heroes come from the same guild print **“Heroes from the same guild can not fight each other.”**

**#Note: There will be no invalid guild name or hero name in this command.**

* **END**
  + When you receive the command end you program should stop accepting commands and print all provinces and the details about them in the format:

**“Province: {province name}”**

**“#Guilds:”**

**“##Guild: Name: {guild1name}, Power: {guild1power}, Size: {guild1size}”**

**“##Guild: Name: {guild2name}, Power: {guild2power}, Size: {guild2size}”**

**“##Guild: Name: {guildNname}, Power: {guildNpower}, Size: {guildNsize}”**

* guild power should be rounded to **two decimal places**.
* If there are no guilds inside the province just print **“#Guilds: None”**
* Order the output alphabetically by province name.
  + The guilds inside the province should be ordered first **by the guilds total points. If there are guilds with equal points they should be ordered alphabetically.**

#### Input

* The input will come in the form of strings. Theses strings will be commands in the format specified above.
* There will be no invalid commands but invalid information in them is possible.
  + For example, a guild or hero with the provided name might not exist.
* The input sequence ends when you receive the command "**END**".

#### Output

* The output consists of the resulting messages from the operations mentioned above, separated by **“---”**.
* Check input/output section to see examples.

#### Constrains

* **All strings** in the input may consist of **any ASCII character**, except **SPACE** so that the input is easily processed.
* There will be **NO invalid** input lines but **invalid information IS possible!**

|  |  |
| --- | --- |
| **Input** | **Output** |
| CREATE\_PROVINCE: Lactacare  ADD\_GUILD: A  ADD\_GUILD: B  ADD\_HERO: A Dunmer Laribus 120 100 100  ADD\_HERO: B Orc Orcsimus 100 100 80  FIGHT: A Laribus B Orcsimus  END | Created province Lactacare  ---  Added Guild: A  ---  Added Guild: B  ---  Added hero: Laribus  ---  Added hero: Orcsimus  ---  Hero Laribus sliced hero Orcsimus.  ---  Province: Lactacare  #Guilds:  ##Guild: Name: A, Power: 320.00, Size: 1  ##Guild: Name: B, Power: 0.00, Size: 0 |
| ADD\_HERO: G6 Breton Lary 100 240 80  CREATE\_PROVINCE: Derby  CREATE\_PROVINCE: Solo  CREATE\_PROVINCE: Fiji  SELECT: Fiji  SELECT: Derby  ADD\_HERO: G6 Breton Lary 100 240 80  DETAILS: G7  DETAILS: G6 Lary  END | No province selected!  ---  Created province Derby  ---  Created province Solo  ---  Created province Fiji  ---  Province Fiji has already been selected!  ---  Province Derby selected!  ---  Added hero: Lary  ---  Guild [G7] does not exist.  ---  Hero: Lary, Type: [BRETON]  #Stats:  Health: 100  Fatigue: 240  Magicka: 80  ---  Province: Derby  #Guilds:  ##Guild: Name: G6, Power: 292.00, Size: 1  Province: Fiji  #Guilds: None  Province: Solo  #Guilds: None |
| CREATE\_PROVINCE: Lactacare  CREATE\_PROVINCE: Lary  SELECT: Lactacare  ADD\_GUILD: A  ADD\_GUILD: B  ADD\_HERO: A Redguard Laribus 120 100 100  ADD\_HERO: B Orc Orcsimus 100 100 80  REMOVE: B Orcsimus  DETAILS: A Laribus  FIGHT: A Laribus B Orcsimus  END | Created province Lactacare  ---  Created province Lary  ---  Province Lactacare selected!  ---  Added Guild: A  ---  Added Guild: B  ---  Added hero: Laribus  ---  Added hero: Orcsimus  ---  Successfully removed hero [Orcsimus] from guild B  ---  Hero: Laribus, Type: [REDGUARD]  #Stats:  Health: 120  Fatigue: 100  Magicka: 100  #Strength bonuses:  Magic damage: +50.0 offense.  ---  No such hero in this guild!  ---  Province: Lactacare  #Guilds:  ##Guild: Name: A, Power: 341.00, Size: 1  ##Guild: Name: B, Power: 0.00, Size: 0  Province: Lary  #Guilds: None |