**Part 2: Draw the memory map when the program runs [3 points]**

**Explain step by step what happened when the program runs and answer some questions.**

**- What is stored in the static heap, stack, dynamic heap?**

**Answer:**

*Stack: hold data for running method*

* *Stack frame for running method*
* *Reference variables (e.g. item, sc, …)*
* *Primitive variables in method (e.g. int, float, char, …)*
* *Running method (e.g. main(), …)*

*Static heap:* *where class definitions, static variables, and method bytecode are stored, shared across all instances of a class*

* *Class definitions (e.g. Item, Vase, Statue, Painting, …)*
* *Static variables, constants, global primitive variables (e.g. PI, …)*
* *Method definition (e.g. main(), …)*

*Dynamic heap:* *where objects (instances of classes) are created and stored at runtime, allowing for dynamic memory allocation*

* *Objects (e.g. item, sc, …) after new method*
* *Dynamic memory (e.g. shrink or grow)*

**- What are objects in the program?**

**Answer:**

*The objects:*

* *item instance of class Vase extends Item*
* *item instance of class Statue extends Item*
* *item instance of class Painting extends item*

**- What is the item variable storing?**

**Answer:**

* *item instance of class Vase extends Item*
* *Instance variables: price, creator, height, material*
* *Method: constructors, getters, setters, inputVase, outputVase*
* *item instance of class Statue extends Item*
* *Instance variables: price, creator, weight, colour*
* *Method: constructors, getters, setters, inputStatue, outputStatue*
* *item instance of class Painting extends item*
* *Instance variables: price, creator, height, width, isWatercolour, isFramed*
* *Method: constructors, getters, setters, inputPainting, outputPainting*

**- Why must you cast to call the method inputVase()/outputVase()?**

**Answer:**

* *Casting is required to access methods that are specific to the subclass, as they are not recognized by the superclass reference type.*
* *It’s good practice to check the actual type of the object (using* ***instanceof****) before casting to avoid ClassCastException at runtime.*

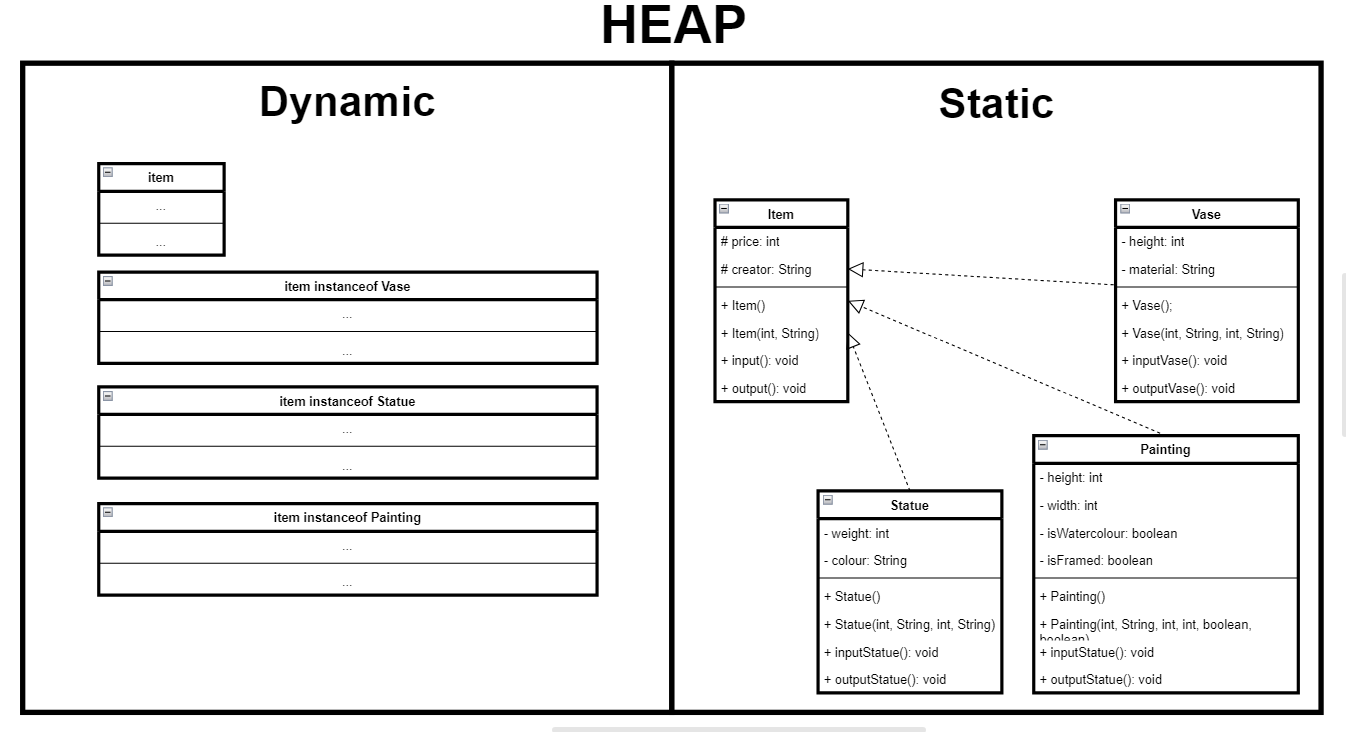
**- What is the error thrown when you cast it wrong?**

**Answer:**

* *ClassCastException*

**- What methods can you call if you don’t cast the item variable?**

**Answer:**

* *Without casting, you can call only those methods that are defined in the Item class. Any subclass-specific methods must be called through a cast to the respective subclass type.*

