

THIS IS WAR



Task 1 Setup

Create a new project environment in your IDE.

Create the following classes:

- Tank.java
- Missile.java
- Main.java

Copy/Paste or Type the Tank and Missile class code into your files.

Task 2 Coding Concepts

An abstraction of a Tank exists in the form of a class.

- i. Explain why the Tank class is considered to be an abstraction of a tank

[2]

--

- ii List the 3 **properties** of the Tank class here:

Property Identifier	Property Datatype

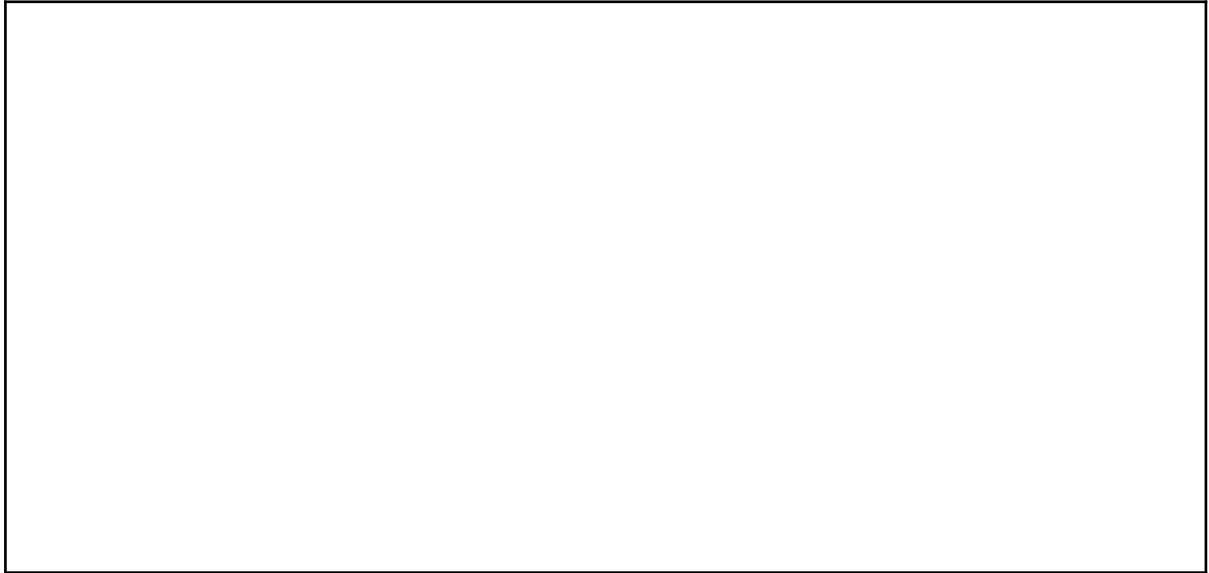
- iii. The Missile class exhibits a form of **polymorphism**. Name this form of **polymorphism** and describe it:

[3]



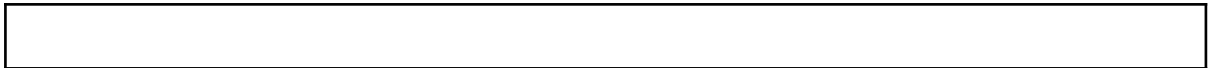
iv. Draw the **UML** diagram describing the relationship between the Tank class and the Missile class. Make the Missile class UML complete but only include the title of the Tank class.

[3]



v. In a Main program, write a line of code to instantiate a new Tank object with the **identifier** exterminator with an id of **ext-1**.

[3]



vi. In Main, add a Missile object to exterminator's Missile array.

[2]



vii. When the Tank class's shootMissile method is called, it:

checks that there is a missile available ie `nextAvailableMissile != -1`
if so it

- Assigns the next available missile to a temporary variable
- decrements `nextAvailableMissile`
- returns the missile reference

else it returns `null`

Code this method.

[5]



viii A tank commander needs to calculate the average damage rating of all currently available missiles in the Tank.

The Tank class method `getAverageRating` calculates and returns the average damage rating of all Missiles in the Tank when the method is called.

Implement the `getAverageRating` method of the Tank class. It returns -1 if there are no Missiles available.

[5]

ix. In a Main program, 2 tanks are attacking each other. At the start of the war both Tanks are loaded with 10 Missiles.

In a loop., the Tanks shoot missiles at each other. If a Tank shoots a missile which has a damage factor greater than 8, then another missile is loaded into the Tank. The first Tank to have no Missiles left is the loser and the result is displayed.

Implement this game code. Use any available Tank and Missile class methods and you can also implement **static** helper methods in the Main program, too.

