**Homework 1- Vildan Tunc**

1. **Why we need to use OOP? Some major OOP languages?**

* The efficiency of the development process increases as code duplication is avoided.
* With the help of the created classes, more work is done with less code and code duplication is prevented.
* Thanks to polymorphism, we get reusable and clean code in the OOP concept.
* Since the created objects are independent of each other, the possibility of hiding information increases.
* Some of the well-known object-oriented languages are Java,Python, C#, C++

1. **Interface vs Abstract class?**

* Abstract classes use inheritance to reduce code duplication.
* We cannot create objects from abstract and interface classes.
* The class extending the abstract class must override all abstract methods of the abstract class.
* And all classes that implement interface class must override all methods.
* While a class can extend only one abstract class, it can implement multiple interface classes.
* We use the extends keyword to use abstract classes, while we use the implements keyword to use interface classes.

1. **Why we need equals and hashcodes? When to override?**

* The equals() and hashcode() are the two important methods provided by the Object class for comparing objects.
* The equals method to compare two objects that whether they are the same, it compares the values of both the object's attributes.
* A hashcode is an integer value associated with every object in Java, facilitating the hashing in hash tables.
* In every class where we override the equals method, we must also override the hashCode method.
* If we do not do this, it will cause our class to work incorrectly when used with hash-based data structures such as HashMap, HashSet, HashTable.

1. **Diamon problem in Java? How to fix it?**

* The diamond problem is a common problem in Java when it comes to inheritance.
* The solution to the diamond problem is default methods and interfaces. We can achieve multiple inheritance by using these two things.

1. **Why we need Garbage Collector? How does it run?**

* Garbage Collector Frees memory and frees up memory space occupied by unused/non-referenced objects.
* It makes java memory efficient because garbage collector removes the unreferenced objects from heap memory.
* It is automatically done by the garbage collector(a part of JVM) so we don't need to make extra efforts.

1. **Java ‘static’ keyword usage?**

* Methods and variables created using the static keyword belong to the class, not objects created from the class.
* Thus, we use it when we want not to create copies of different objects with different functions.
* Only one is created in memory.
* We can create it using the static keyword, or we can create it with static blocks.
* Therefore, every object that will use it reaches the only address allocated to it.

1. **Immutability means? Where, How and Why to use it?**

* An immutable object is an object whose internal state remains constant after it has been entirely created.
* If we don't want a variable or method we created to be changed, we can make it immutable using the final keyword.
* Strings in Java are also immutable. Their values do not change after they are created.
* String Pool is possible in Java because String is immutable.

1. **Composition and Aggregation means and differences?**

* Composition(mixture) is a way to wrap simple objects or data types into a single unit.
* Aggregation(collection) differs from ordinary composition in that it does not imply ownership.
* While Composition is a strong association, aggregation is a weak association.

1. **Cohesion and Coupling means and differences?**

* Cohesion: While the relationship between classes is desired to be loosely coupled, procedures and data fields within classes are desired to be cohesive/cohesive.
* A class should perform only one task within the framework of the Single responsibility principle.
* Coupling: It represents the relationship between more than one class/object.
* while Highly cohesive gives the best software, Where as loosely coupling gives the best software.

1. **Heap and Stack means and differences**

* The JVM divides the memory into two parts: stack memory and heap memory.
* While items that have a very short life such as methods, variables, and reference variables of the objects are stored in stack,
* objects and Java Runtime Environment (JRE) classes are stored in heap.
* While Stack follows the LIFO order, Heap does not follow any order because it is a dynamic memory allocation.
* While Stack has faster Access, allocation, Heap has slower Access, allocation
* Stack is smaller in size and not flexible, but Heap is larger in size and flexible

1. **Exception means? Type of Exceptions?**

* Exceptions are exceptions that occur during program runtime.
* In Java, all errors are represented by classes, an error object is created when an error occurs.
* If the exception created in a method is sent to the expression that calls the method, we can call it throwing an exception.
* When an error is thrown, an exception is handled in the code block. This is called exception handling.
* There are two type of exceptions which are :
* **Checked Exception:** We need to handle the error throwing status of some code blocks. For example, when we want to access a file, we should handle the exception in case there is no file.
* **Runtime exception(unchecked exception) :** These are the error types that are not required to be handled by the compiler. An example is the ArithmeticException it throws when we are trying to divide a value by 0.
* And there is also an **Error**. It occurs when serious errors occur in the program. Like
* OutOfMemoryError, StackOverFlowError

1. **How to summarize ‘clean code’ as short as possible?**

* The fact that the code is simply readable and understandable, that new features can be added without changing the current code, that it is easy to maintain, that it is expandable, shows that it is clean code.

1. **What is the method of hiding in Java?**

* If a subclass defines a static method with the same signature as a static method in the super class, in such a case, the method in the subclass hides the one in the superclass.
* Both methods must be static.

1. **What is the differences between abstraction and polymorphism in Java?**

* Abstraction is that Abstraction is implemented using abstract class and interface in Java while Polymorphism is supported by overloading and overriding in Java.
* Abstraction allows a programmer to design software better by thinking in general terms rather than specific terms while Polymorphism allows a programmer to defer choosing the code you want to execute at runtime.