**Pattern problems:**

**1)**

1 2 3 4 5

6 7 8 9

10 11 12

13 14

15

**2)**

3

6 9

12 15 18

21 24 27 30

**Java is Object Oriented Programming System – (OOPS)**

**Structure:**

* **Class -  A class is**a user defined blueprint or prototype from which objects are created**.**
* **Object – Instances of class**
* **Method – it is member of a class which contains set of java statements. It has a return types, access modifier and can have attributes**
* **Attributes – Parameters passed inside a method are called attributes**

**Principals of OOP**

* **Inheritance**
* **Abstraction**
* **Polymorphism**
* **Encapsulation**

**Inheritance : Acquiring properties from parent**

**Extends keyword is used to get properties from parent to child.**

**Parent class – Vehicle - KTM:**

**petrolEngine**

**desielEngine**

**EV**

**gears**

**braking**

**Child RC extends Vehicle - KTM -**

**Type**

**Colour**

**Style**

**Child Duke**

**Type**

**Colour**

**Style**

**Types of Inheritance:**

1. **Single inheritance – only one parent and one child**
2. **Multilevel inheritance –**
3. **Hierarchical inheritance - one parent class is inherited by more than one child class**
4. **Multiple Inheritance – not possible in java ( Interface helps us to achieve this)**
5. **Hybrid Inheritance - not possible in java (Hierarchical + Multiple) ( is not possible because it leads to diamond problem)**

**Interface:**

1. Class declaration contains Interface keyword
2. All the methods as by default abstract
3. By default the access modifier of all the methods and variables are public
4. Methods should not be private or protected

**Abstraction:**

1. Class declaration contains Abstract keyword
2. Not all methods are abstract/ atleast one methods should be concrete
3. Methods should not be private

Is it possible to extend an abstract class in to a child class without implementing its methods

A: Yes, it is possible but for that we have to make the child class also as abstract

Abstraction: Hiding implementation from the user( only providing him the necessary details like how to use but not how it is happening in backend(implementation/code))

Polymosphism

Overloading :

[**sendKeys() can have multiple parameters**](https://seleniumtipsandtricks.quora.com/sendKeys-can-have-multiple-parameters)

If you ever type in a textbox using this code

1. element.sendKeys(“java”);
2. element.sendKeys(Keys.ENTER);
3. element.sendKeys(Keys.TAB);

you can replace it with

1. element.sendKeys(“java”, Keys.ENTER, Keys.TAB);

Overridding

Polymorphism means "**many forms**",

**Method Overloading:**

If there are multiple methods in a class and those have same name with different attributes or data types. Such methods are called overloaded or this kind of declaration is called over loading

Method overloading can only be done in same class

**Method Overriding:**

A method from parent class is Overridden in child class by just changing it’s definition or body. This achieved by inheritance. That means we should have at least two classes to perform overriding.

1 2 3

4 5 6

7 -10 9

One and multidimensional array

Date concepts

Simple date format

Calendar

Constructors:

1. Constructor’s name should be same as class
2. It does not return any value. It doesn’t have a return type
3. It is block of code or a special method of class which will be called automatically when object is created for that class.
4. To instantiate variables of a class

Constructors:

Implicit Constructors

Explicit Constructors:

1. no-arg Constructors

2.Parametrised constructors

Important Keywords:

**Super-**

Calling parent variable and parent method;

To send the properties from child class to parent class

**This**

**Instance variables**

**Local Variables**

**Class variables**

**Static keyword**

**Static variables and methods**

**Static blocks**

**Final keyword:**

* We cannot inherit a class which Is declared as final
* We cannot override a method from parent class in child class if the parent class method is declared ad final
* We cannot reassign/change value of a variable if is declared as final.

Packages:

A package is collection of classes and interfaces.

Default package – java.lang is by default available in java compiler

User defined packages – define by user

Java.util = collection interfaces

Package syntax

Import packageName/Path.ClassName;

Importance is

If two classes are in same package, then those classes can interact directly.

If the classes are in different packages. And if you wan to interact with each other. Then you have to import the package

Eg:

* 1. You have to make a payment with debit card – import debitCard.payment
  2. Payment by coupons - import coupons.payment
  3. You have to make payment with credit card ---import CreditCard.payment

**Access specifiers/modifiers**

**Public, private, protected, default**

**Public: Public methods/variables can be accessible anywhere in the project**

**Default: default methods/variables can be accessible only inside same package**

**Private: private methods/variables can be accessible only inside same class**

**Protected: protected methods/variables can be accessible in subclass if that subclass is in different package.**

**Exceptions:**

**Try block is used to handle exception. All the statements present after try block are executed though exception occurs or not.**

**Always try block should be followed by either catch or finally**

**Arrays:** Java array is **an object which contains elements of a similar data type**.

**Java Collections Framework:**

Java **Collections Framework objects can contain elements of a different data type**.

**Is a collection of interfaces and classes which helps in storing and processing the data efficiently.**

**Collections objection contains elements of same or different datatypes. We can make collections object to store strictly single data types**

**Collection has three types processes or classes:**

* **List - interface**
  + **ArrayList - discussed**
  + **LinkedList – please do research**
  + **Vector - please do research**
* **Set**
  + **HashSet - discussed**
  + **LinkedHashSet - please do research**
  + **TreeSet - please do research**
* **Map**
  + **HashMap**
  + **Hash Table**
  + **TreeMap**

**Difference between Hash Map and HahTable**

1. **Synchronization/Thread safe**
   1. **HashMap – not synchronised**
   2. **HashTable – synchronized**
2. **Null keys & Vaalues**
   1. **HashMap – accepts null keys and values**
   2. **HashTable – does not accepts null keys and values**
3. **Iterating the values**
   1. **Hashmap - Iterator**
   2. **HashTable - Enumerator**

**List**

**ArrayList:**

**We use indexes to point or retrieve data from list**

**It is a ordered list.**

**It allows duplicates**

**Set:**

**We do not have indexes. We use iterator.**

**It is un-ordered list**

**It does not accept duplicate values.**