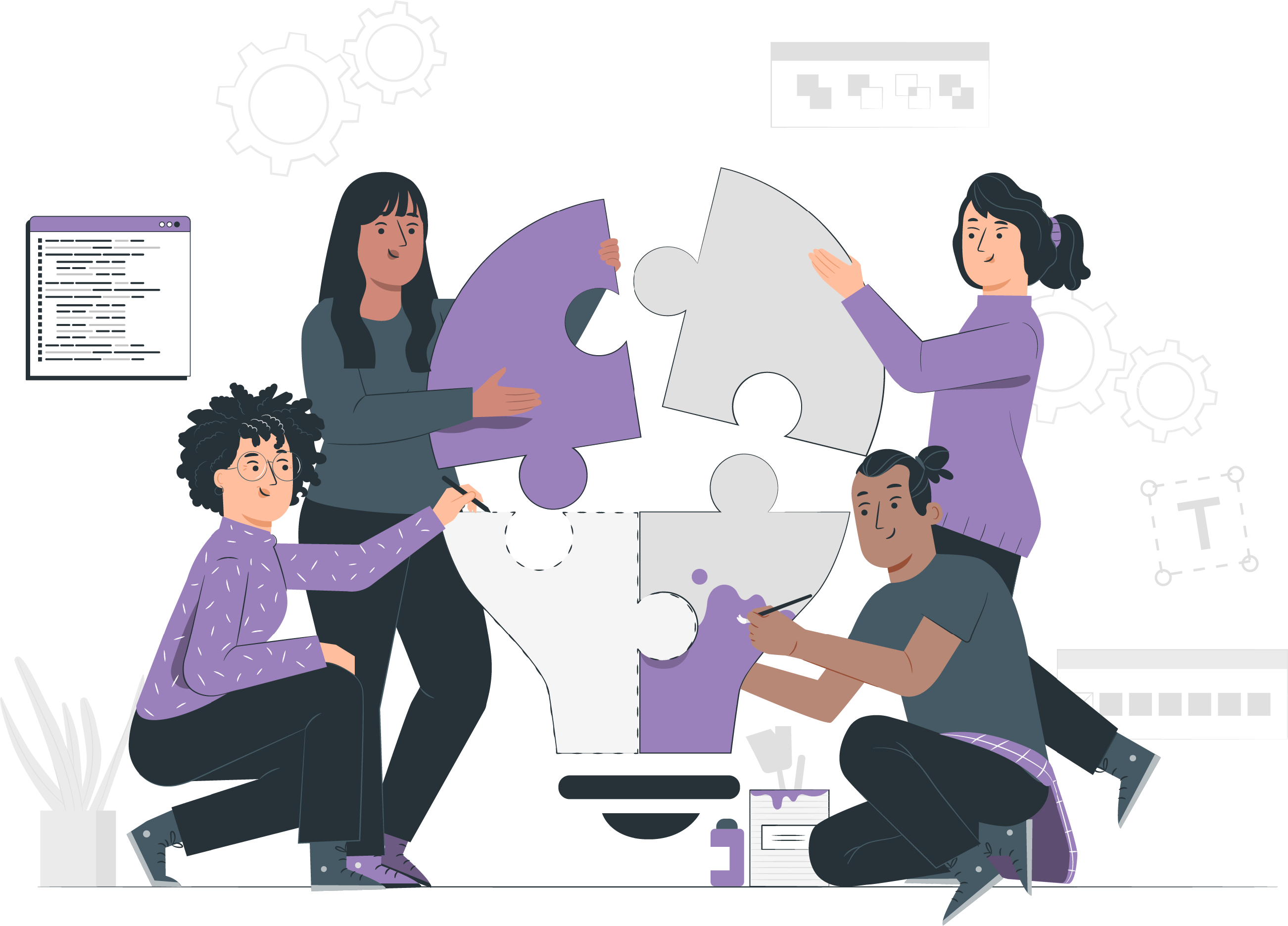
**1. Why do we need static keyword in Java Explain with an example?**



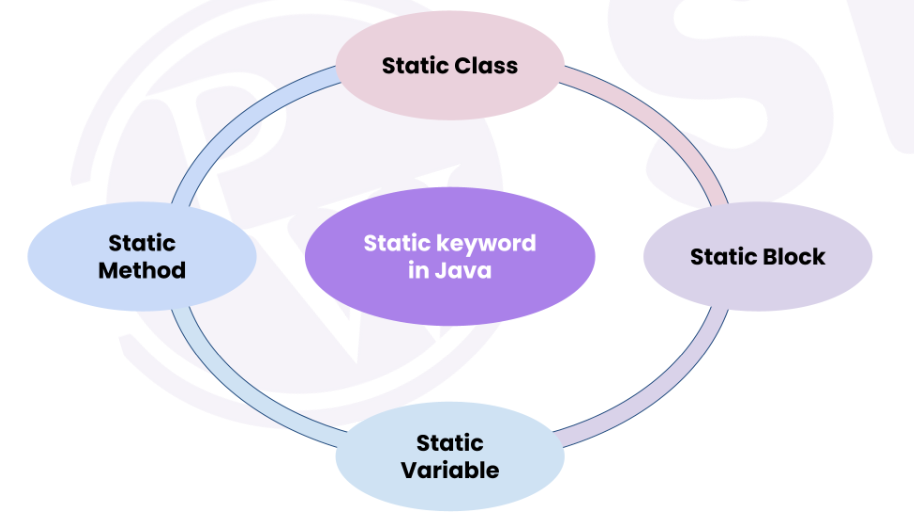
=> The static keyword is mainly used for memory management in Java. A static keyword can be applied to variables, blocks, methods, and classes. The static keyword is a property of a class rather than an instance of the class. The static keyword is used for a constant variable or a method that is the same for every instance of a class.

The static keyword is a non-access modifier in Java and is applicable for the following:

1. Variables

2. Methods

3. Blocks   
4. Class



***Example***

class Student

{

int age = 5; // instance variable, non - static variable

static String school; // non - instance variable, static variable

static

{

school = "PW";

}

public void show()

{

System.out.println("in show " + age + " " + school); // static variable can be accessed in non-static method

}

public static void study()

{

System.out.println("studying "); // non-static variable cannot be accessed here

}

}

public class Demo

{

static{

System.out.println("in static block");

}

public static void main(String[] args) {

System.out.println("in main method");

Student obj = new Student();

Student.study();

obj.age = 19;

obj.school = "PWSkills";

Student obj1 = new Student();

System.out.println(obj1.school);

obj.show(); // in show 19 PWSkills

obj1.show(); // in show 0 ?

}

}

**2. What is class loading and how does the Java program actually executes?**

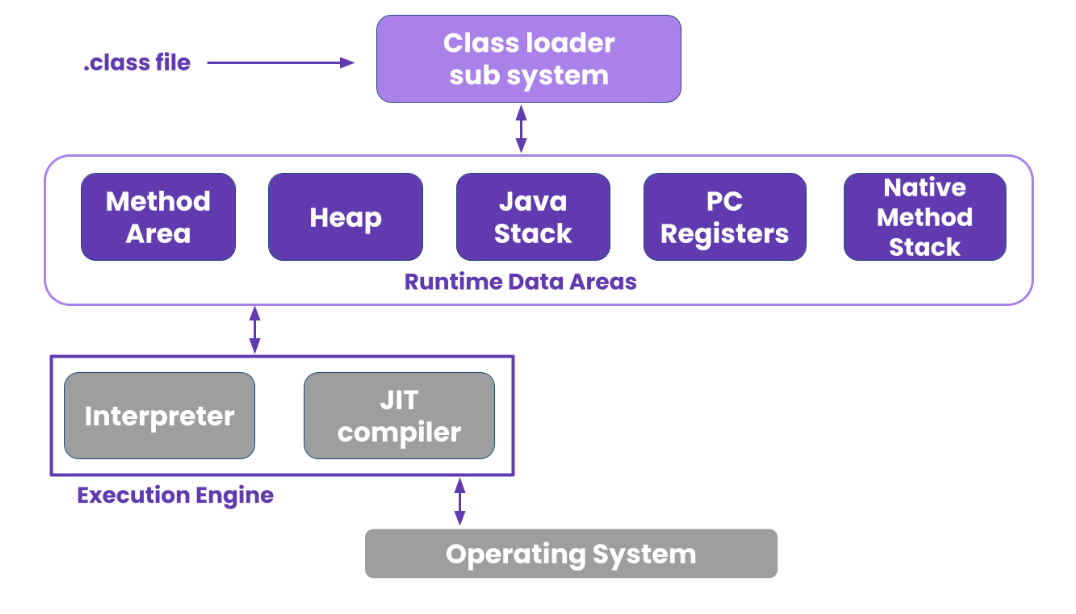
=> In Java, classloading is the process of loading class files into the JVM (Java Virtual Machine) at runtime. It is responsible for loading classes from various sources, such as the file system, network, and databases, and making them available to the JVM for execution.

The class loading process in Java is divided into three phases: loading, linking, and initialization.

**1.** **Loading:** In the loading phase, the classloader locates the class file using the fully qualified class name, reads the class file, and converts it into a Class object. The Class object contains the metadata of the class, such as the fields, methods, and constructors.

**2.** **Linking:** In the linking phase, the JVM performs several operations on the Class object, such as verifying the class file's integrity, resolving symbolic references, and allocating memory for the class variables.

**3. Initialization:** In the initialization phase, the JVM initializes the class variables with their default values, and runs the class's static initialization block (if any).



**3.Can we mark a local variable as static?**

=>No, A static filed/variable belongs to the class and it will be loaded into the memory along with the class. You can invoke them without creating an object. (using the class name as reference). There is only one copy of the static field available throughout the class i.e. the value of the static field will be same in all objects.

In Java, a static variable is a class variable (for whole class). So if we have static local variable (a variable with scope limited to function), it violates the purpose of static. Hence compiler does not allow static local variable

**4.Why is the static block executed before the main method in java?**

=> The static blocks always execute first before the main() method in Java because the compiler stores them in memory at the time of class loading and before the object creation.

***Example:***

public class Demo {  
 static {  
 System.*out*.println("Static");  
 }  
 public static void main(String[] args) {  
 System.*out*.println("Main()");

}  
}

***Output:***

Static

Main()

**5. Why is a static method also called a class method?**

=> A static method is a method that belongs to a class rather than an instance of a class. This means you can call a static method without creating an object of the class. That’s why Static methods are sometimes called class methods.

**6. What is the use of static blocks in java?**

=> It is used to initialize static data members. It is used to initialize before the main method at the time of class loading. It gets executed only once when the class gets loaded. It is not necessary to execute it again when creating different objects after the first time.

* If we want to perform any activity at the time of loading a .class file we have to define that activity inside the static block.
* We can write any no of static blocks, those static blocks will be executed from top to bottom.
* Normally a static block is used to perform initialization of the static variables.

**7. Difference between Static and Instance member.**

=> **Instance Member:**

An instance member is essentially anything within a class that is not marked as static. That is, that it can only be used after an instance of the class has been made (with the new keyword). This is because instance members belong to the object, whereas static members belong to the class.

**Static Member:**

Static members are those which belong to the class and you can access these members without instantiating the class. The static keyword can be used with methods, fields, classes (inner/nested), blocks.

**8.Difference between static and non static variable .**

=> **Static variable**

* These variables are called “class variables”.
* These variables will get memory in the method area.
* If the value does not change from object to object then we need to use static variables.
* Inside a static area we can access static variables only.
* Static variables are created using static keywords.

**Non-static variable**

* These variables are called “instance variables”.
* These variables will get memory in the heap area.
* If the value changes from object to object then we need to use “non-static” variables.
* Inside a nonstatic area we can access both static and non-static variable.
* Non-static variables are created without using the “static” keyword.