ASSIGNMENT 1 ON JAVA GENERICS

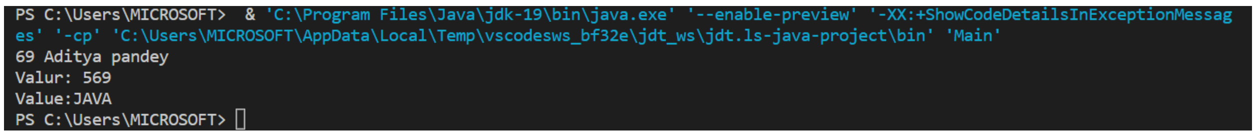
1) AIM: Write a Java Program to demonstrate a Generic Class.

THEORY: We can create a class that can be used with any type of data. Such a class is known as Generics Class. The Java Generics allows us to create a single class, interface, and method that can be used with different types of data (objects).This helps us to reuse our code.

PROGRAM:

```java
class GenericClass<T> {

   private T data;
   public GenericClass(T data) {
      this.data = data;
   }
   public T getData() {
      return this.data;
   }
}
public class aditya1 {

   public static void main(String[] args) {
      // TODO Auto-generated method stub
      GenericClass<Integer> intObj = new GenericClass<>(569);
      System.out.println("69 Aditya Pandey");
      System.out.println("Value:"+ intObj.getData());
      GenericClass<String> StrObj = new GenericClass<>("Java");
      System.out.println("Value:"+ StrObj.getData());
   }

}
```

OUTPUT:



```
PS C:\Users\MICROSOFT>  & 'C:\Program Files\Java\jdk-19\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessag
es' '-cp' 'C:\Users\MICROSOFT\AppData\Local\Temp\vscodesws_bf32e\jdt_ws\jdt.ls-java-project\bin' 'Main'
69 Aditya pandey
Valur: 569
Value:JAVA
PS C:\Users\MICROSOFT> 
```

CONCLUSION: Hence, we understood the concept of generics class.

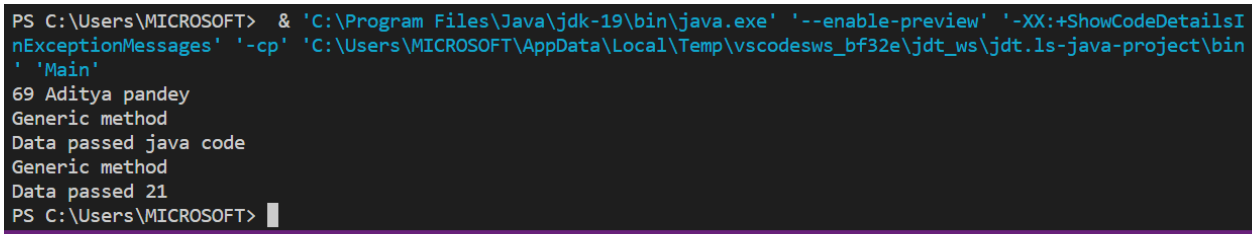
2) AIM: Write a Java Program to demonstrate a Generic Class.

THEORY: Similar to the generics class, we can also create a method that can be used with any type of data. Such a class is known as Generics Method.

PROGRAM:

```java
class DemoClass{
   public<T> void genericMethod(T data) {
      System.out.println("Generic Method");
      System.out.println("Data passed "+ data);
   }
}

public class aditya {

   public static void main(String[] args) {
      // TODO Auto-generated method stub
      DemoClass demo = new DemoClass();
      System.out.println("69 Aditya Pandey");
      demo.<String>genericMethod("java code");
      demo.<Integer>genericMethod(21);

   }
}
```

OUTPUT:



```
PS C:\Users\MICROSOFT>  & 'C:\Program Files\Java\jdk-19\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsI
nExceptionMessages' '-cp' 'C:\Users\MICROSOFT\AppData\Local\Temp\vscodesws_bf32e\jdt_ws\jdt.ls-java-project\bin
' 'Main'
69 Aditya pandey
Generic method
Data passed java code
Generic method
Data passed 21
PS C:\Users\MICROSOFT>
```

CONCLUSION: Hence, we understood the concept of generics method.

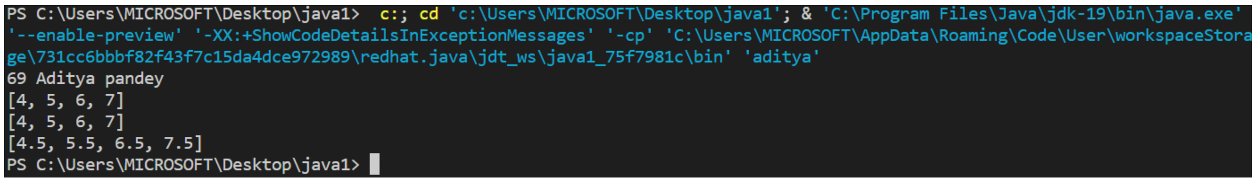
3) AIM: Write a Java Program to demonstrate Wildcards in Java Generics

THEORY: The question mark (?) is known as the wildcard in generic programming. It represents an unknown type. The wildcard can be used in a variety of situations such as the type of a parameter, field, or local variable; sometimes as a return type. Unlike arrays, different instantiations of a generic type are not compatible with each other, not even explicitly. This incompatibility may be softened by the wildcard if ? is used as an actual type parameter.

PROGRAM:

```java
import java.util.Arrays;
import java.util.List;

public class aditya{
  static void displayUpperBounds(List<? extends Number>list) {
    System.out.println(list);
  }

static void displayLowerBounds(List<? super Integer>list) {
  System.out.println(list);
}

static void displayUnBounds(List<?>list) {
  System.out.println(list);
}

public static void main(String[] args) {
  // TODO Auto-generated method stub
  System.out.println("69 Aditya pandey");
  List<Integer> list1 = Arrays.asList(4,5,6,7);
  displayUpperBounds(list1);
  List<Number> list2 = Arrays.asList(4,5,6,7);
  displayLowerBounds(list2);
  List<Double> list3 = Arrays.asList(4.5,5.5,6.5,7.5);
  displayUnBounds(list3);
  }
}
```

OUTPUT:



```
PS C:\Users\MICROSOFT\Desktop\java1>  c:; cd 'c:\Users\MICROSOFT\Desktop\java1'; & 'C:\Program Files\Java\jdk-19\bin\java.exe'
'--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\MICROSOFT\AppData\Roaming\Code\User\workspaceStora
ge\731cc6bbbf82f43f7c15da4dce972989\redhat.java\jdt_ws\java1_75f7981c\bin' 'aditya'
69 Aditya pandey
[4, 5, 6, 7]
[4, 5, 6, 7]
[4.5, 5.5, 6.5, 7.5]
PS C:\Users\MICROSOFT\Desktop\java1> 
```

CONCLUSION: Hence, we understood the concept of Wildcards in Java Generics.