Programs based on Java Generics

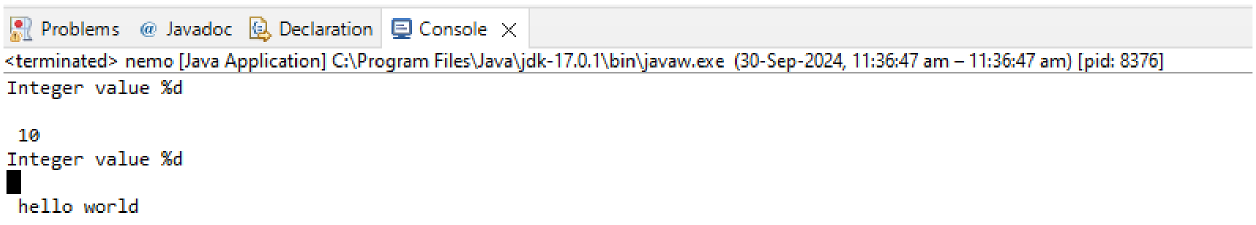
Practical 1

Program no 1:
Aim: Write a java program to demonstrate generic class
Code:

```
import java.io.*;
import java.util.*;

public class Box<T>
{
	private T t;
	public void add(T t)
	{
		this.t = t;
	}
	public T get()
	{
		return t;
	}
	public static void main(String [] args)
	{
		Box <Integer> integerBox = new Box <Integer>();
		Box <String> stringBox = new Box <String>();
		integerBox.add(new ~~Integer~~ (10));
		stringBox.add(new String("Hello world"));
		System.out.printf("Integer Value: %d\n\n", integerBox.get());
		System.out.printf("String Value: %s\n", stringBox.get());
	}
```

Output:



```
<terminated> nemo [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe  (30-Sep-2024, 11:36:47 am – 11:36:47 am) [pid: 8376]
Integer value %d

 10
Integer value %d

 hello world
```

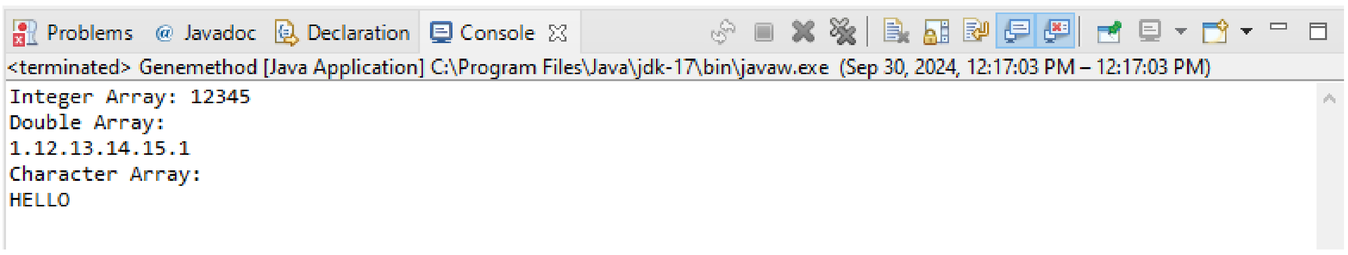
Programs based on Java Generics

Program no 2:
Aim: Write a Java program to Demonstrate Generic Method.
Code:

```
import java.io.*;
import java.util.*;

public class Genemethod
{
        public static<E> void printArray(E[]arr)
        {
                for(E element:arr)
                {
                        System.out.printf("%S", element);
                }
                System.out.println();
        }
        public static void main(String[]args)
        {
                Integer[] intarr = {1,2,3,4,5};
                Double[] darr = {1.1,2.1,3.1,4.1,5.1};
                Character[] carr = {'H','e','l','l','o'};
                System.out.printf("Integer Array: ");
                printArray(intarr);
                System.out.println("Double Array: ");
                printArray(darr);
                System.out.println("Character Array: ");
                printArray(carr);
        }
}
```

Output:



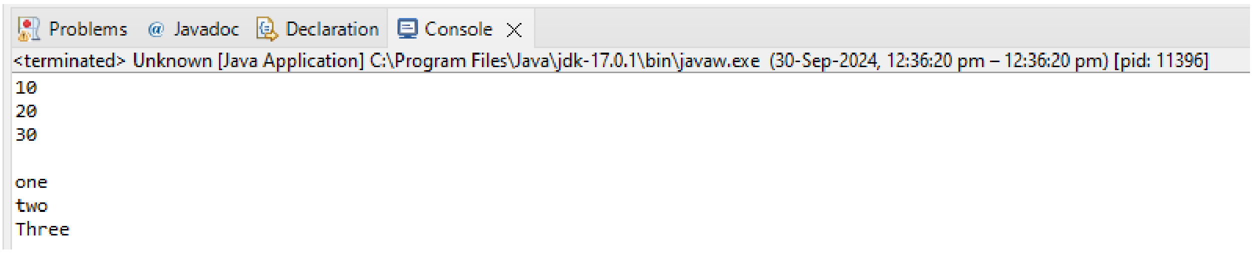
```
<terminated> Genemethod [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (Sep 30, 2024, 12:17:03 PM – 12:17:03 PM)
Integer Array: 12345
Double Array:
1.12.13.14.15.1
Character Array:
HELLO
```

Program no 3:
Aim: Write a Java program to Demonstrate Generic Method.
Code:

```
import java.io.*;
import java.util.*;
public class Unknown
{
	static void processElements(ArrayList<?>a)
	{
		for(Object element:a)
		{
			System.out.println(element);
		}
	}
	public static void main(String[]args)
	{
		ArrayList<Integer> a1 = new ArrayList<>();
		a1.add(10);
		a1.add(20);
		a1.add(30);
		processElements(a1);

		ArrayList<String> a2 = new ArrayList<>();
		a2.add("one");
		a2.add("Two");
		a2.add("Three");
		processElements(a2);
	}
}
```

Output:



```
<terminated> Unknown [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe  (30-Sep-2024, 12:36:20 pm – 12:36:20 pm) [pid: 11396]
10
20
30

one
two
Three
```

Programs based on Java Generics

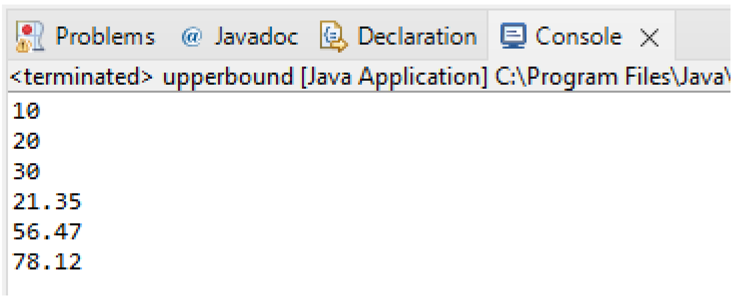
Program no 4:
Aim: Write a Java program to Demonstrate Wildcard arguments with an Upper Bound.
Code:

```
package mca44;
import java.util.*;

public class upperbound {
	static void processElement(ArrayList<? extends Number> a)
	{
		for (Object element:a)
		{
			System.out.println(element);
		}
	}
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		ArrayList<Integer> a1=new ArrayList<>();
		a1.add(10);
		a1.add(20);
		a1.add(30);
		processElement(a1);
		ArrayList<Double> a2=new ArrayList<>();
		a2.add(21.35);
		a2.add(56.47);
		a2.add(78.12);
		processElement(a2);

	}
}
```

Output:



```
<terminated> upperbound [Java Application] C:\Program Files\Java\
10
20
30
21.35
56.47
78.12
```

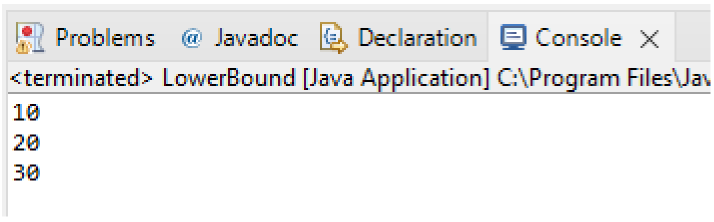
Programs based on Java Generics

Program no 5:
Aim: Write a Java program to Demonstrate Wildcard arguments with an Lower Bound.
Code:

```
package mca44;
import java.util.*;

public class LowerBound
{
	static void processElement(ArrayList<? super Integer> a)
	{
		for (Object element:a)
		{
			System.out.println(element);
		}
	}
	public static void main(String[] args)
	{
		// TODO Auto-generated method stub
		ArrayList<Integer> a1=new ArrayList<>();
		a1.add(10);
		a1.add(20);
		a1.add(30);
		processElement(a1);
		ArrayList<Double> a2=new ArrayList<>();
		a2.add(21.35);
		a2.add(56.47);
		a2.add(78.12);
	}
}
```

Output:



```
<terminated> LowerBound [Java Application] C:\Program Files\Jav
10
20
30
```