PRACTICAL - 2

Running Java code on android studio and Demonstrating Activity Lifecycle

1. Write down steps to run java program in android studio

- From the project folder select java folder
- Right click on java folder and select new then java class.
- Provide a class name and press ok.
- Right the java code.
- Right click on java file and select run “Print.main()” with coverage**.**

2. Write a java program to print 1 to 10 using for loop.

Program1.java

```
package com.example.firstapp;

public class program1 {
   public static void main(String[] args) {
      for (int i = 1; i <= 10; ++i) {
         System.out.println(i);
      }
   }
```

} **Output**:

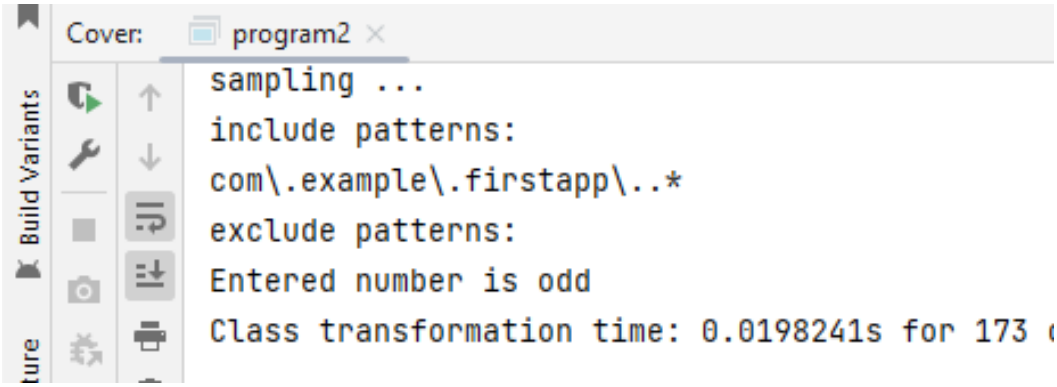
PRACTICAL - 2

3. Write a java program to check whether the entered number is odd or even.

Program2.java

```
package com.example.firstapp;

public class program2 {
    public static void main(String[] args) {
        int num = 5;
        if (num % 2 == 0)
            System.out.println("Entered number is
even");
        else
            System.out.println("Entered number is
odd");
    }
}
```

Output:



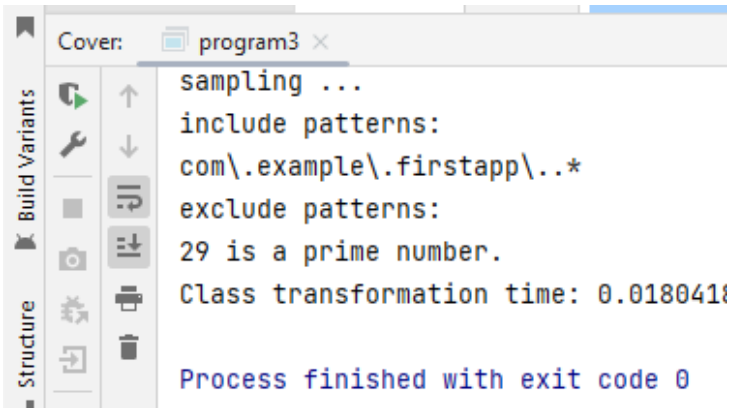
4. Write a java program to check whether the entered number is prime or not.

Program3.java

```
package com.example.firstapp;

public class program3 {
    public static void main(String[] args) {
        int num = 29;
        boolean isprimenum = true;
```

```
        for (int i = 2; i <= num / 2; ++i) {
            if (num % i == 0) {
                isprimenum = false;
                break;
            }
        }
        if (isprimenum)
            System.out.println(num + " is a prime
number.");
        else
            System.out.println(num + " is not a prime
number.");
    }
}
```

Output:



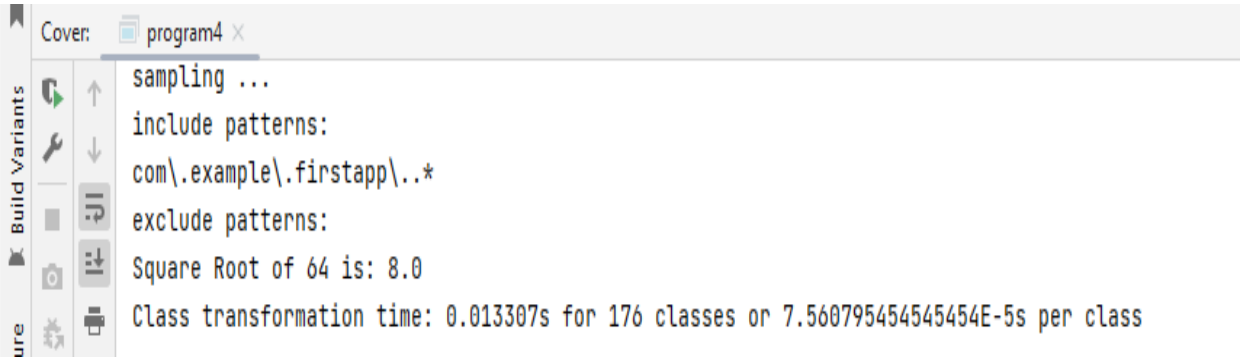
5. Write a Java Program to Find Square Root of a Number Without sqrt Method.

Program4.java

```
package com.example.firstapp;

public class program4 {
    public static void main(String[] args) {
        int number = 64;
        double temp;

        double squareRoot = number / 2;
```

```
        do {
            temp = squareRoot;
            squareRoot = (temp + (number / temp)) / 2;
        } while ((temp - squareRoot) != 0);

        System.out.println("Square Root of "+ number+ "
is: "+squareRoot);
    }
}
```

Output:



6. Write a Java Program to Display Even Numbers From 1 to 100.

Program5.java

```
package com.example.firstapp;
public class program5 {
    public static void main(String[] args) {
        int limit = 100;
        System.out.println("Printing Even numbers
between 1 and " + limit);
        for(int i=1; i <= limit; i++){
            if( i % 2 == 0){
                System.out.print(i + " ");
            }
        }
        System.out.println();
    }
}
```

Output:

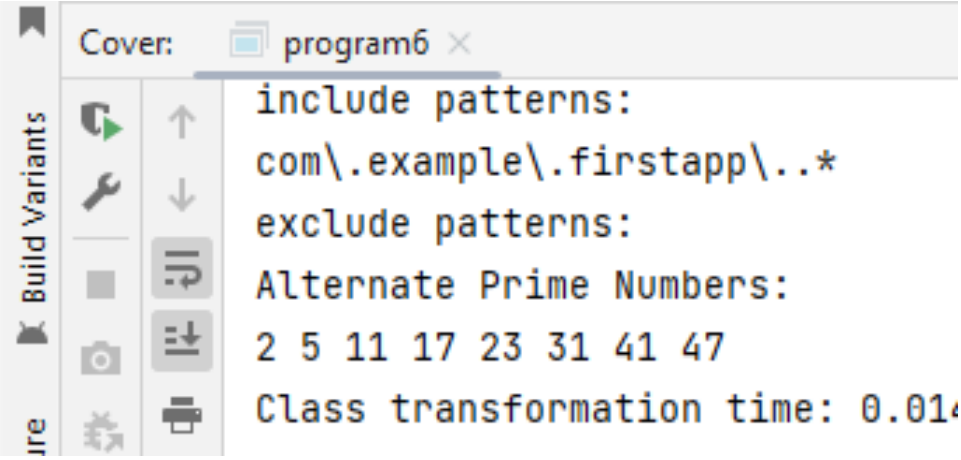
```
include patterns:
com\.example\.firstapp\..*
exclude patterns:
Printing Even numbers between 1 and 100
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100
Class transformation time: 0.0158707s for 185 classes or 8.578756756756758E-5s per class
```

7. Write a Java Program to Display Alternate Prime Numbers.

Program6.java

```
package com.example.firstapp;
public class program6 {
    public static boolean isPrime(int number) {
        if (number <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false;
            }
        }
        return true;
    }
    public static void main(String[] args) {
        System.out.println("Alternate Prime Numbers:");
        int count = 0;
        for (int i = 1; i <= 50; i++) {
            if (isPrime(i) && count % 2 == 0) {
                System.out.print(i + " ");
            }
            if (isPrime(i)) {
                count++;
            }
        }
        System.out.println();
    }
}
```

Output:

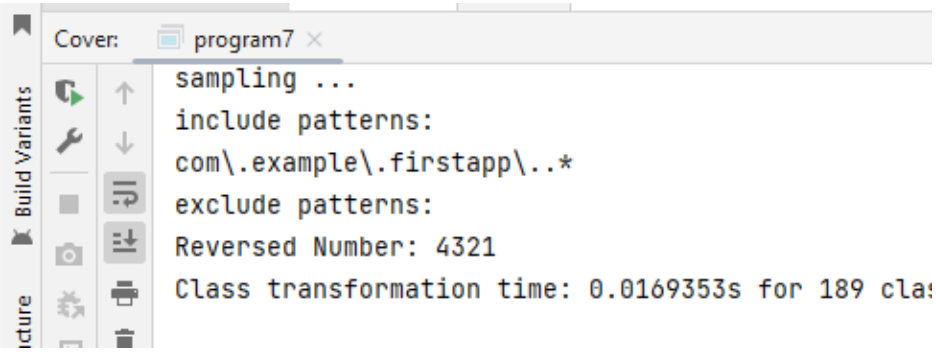


8. Write a Java Program to Reverse a Number.

Program7.java

```
package com.example.firstapp;

public class program7 {
    public static void main(String[] args) {
        int num = 1234, reversed = 0;
        while(num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }
        System.out.println("Reversed Number: " +
reversed);
    }
}
```

Output:



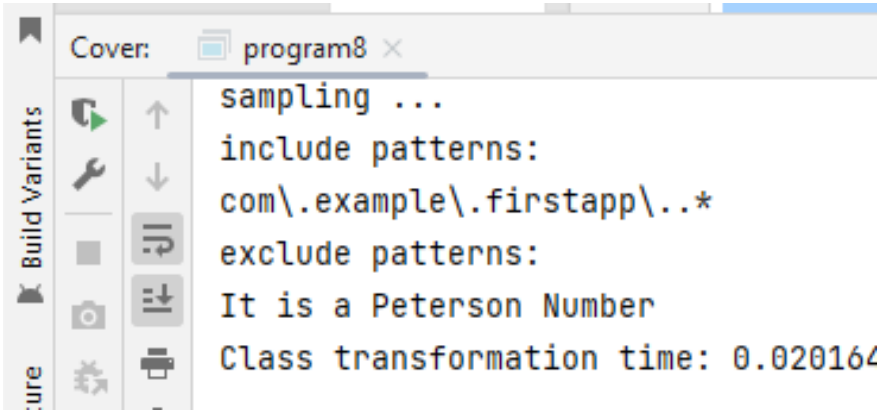
PRACTICAL - 2

9. Write a Java Program to check whether the entered number is a Peterson Number or not.

Program8.java

```
package com.example.firstapp;

public class program8 {
    public static void main(String[] args) {
        int num = 145;
        int sum = 0;
        int temp = num;
        while (temp > 0) {
            int rem = temp % 10;
            int fact = 1;
            for (int i = 1; i <= rem; i++) {
                fact = fact * i;
            }
            sum = sum + fact;
            temp = temp / 10;
        }
        if (sum == num) {
            System.out.println("It is a Peterson
Number");
        } else {
            System.out.println("It is not a Peterson
Number");
        }
    }
}
```

Output:

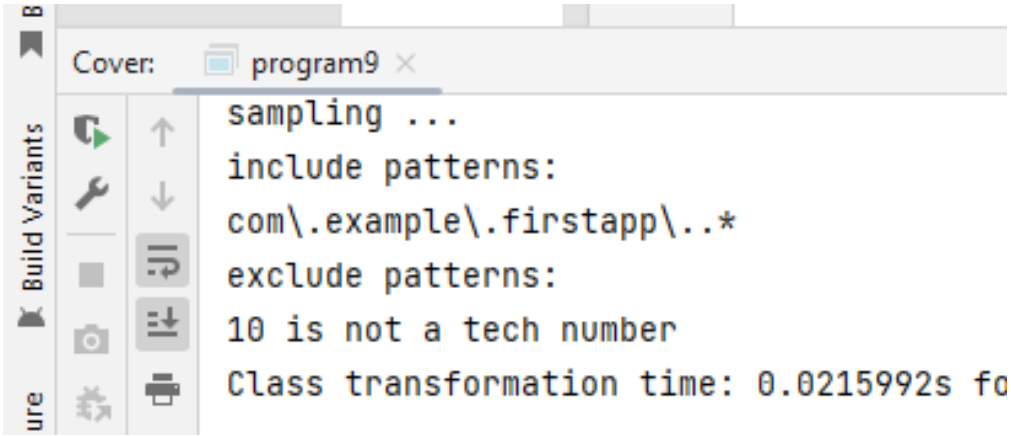


PRACTICAL - 2

10. Write a Java Program to check whether the entered number is a Tech Number or not.

```
package com.example.firstapp;

public class program9 {
    public static void main(String[] args) {
        int num = 10;
        int sum = 0;
        int prod = 1;
        int temp = num;
        while (num > 0) {
            int rem = num % 10;
            sum = sum + rem;
            prod = prod * rem;
            num = num / 10;
        }
        if (sum == prod) {
            System.out.println(temp + " is a tech
number");
        } else {
            System.out.println(temp + " is not a tech
number");
        }
    }
}
```

Output:



Demonstrating Activity Life Cycle

MainActivity.java

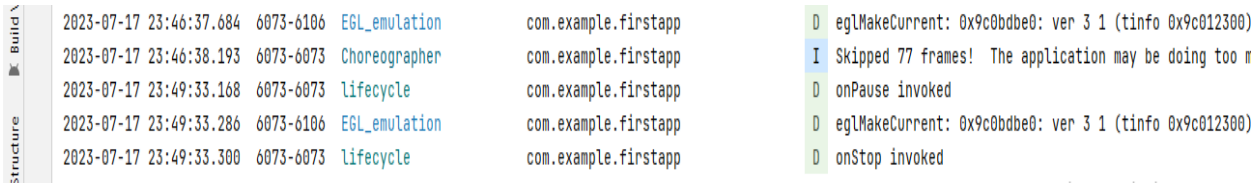
```
package com.example.firstapp;
import android.util.Log;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState)
{
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("lifecycle","onCreate invoked");
    }
    @Override
    protected void onStart() {
        super.onStart();
        Log.d("lifecycle","onStart invoked");
    }
    @Override
    protected void onResume() {
        super.onResume();
        Log.d("lifecycle","onResume invoked");
    }
    @Override
    protected void onPause() {
        super.onPause();
        Log.d("lifecycle","onPause invoked");
    }
    @Override
    protected void onStop() {
        super.onStop();
        Log.d("lifecycle","onStop invoked");
    }
    @Override
    protected void onRestart() {
        super.onRestart();
```

```
        Log.d("lifecycle","onRestart invoked");
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d("lifecycle","onDestroy invoked");
    }
}
```

Output:

Now see on the logcat: onCreate, onStart and onResume methods are invoked.

Now click on the HOME Button. You will see onPause method is invoked. After a while, you will see onStop method is invoked.



Now see on the emulator. It is on the home. Now click on the Center button to launch the app again.