Q1

```
package java11;


	interface Interest{
	   double calc(double a,double b,int c);
	}


	public class Simpleintrest{
	   public static void main(String[] args) {


	      var F=600000;
	      var rate=7;
	      var time=4;


	      Interest i=(var a, var b, var c)->(a*b*c)/100;
	      System.out.println(i.calc(F,rate,time));


	   }
	}
```

---

q2

```
package java11;
import java.util.ArrayList;


public class Q2var {
```

```
//    var x=50;
// var cannot be used in an instance and global variable declaration

   public static void main(String[] args) {
       var x =50;         //this is acceptable
//       var s;             //var cannot be used without explicit initialization


//     var<var> l1=new ArrayList<>();
//     We need to Specify Type, var cannot be used as a Generic type


//       var<Integer> l2=new ArrayList<>();
//       Even if generic type is specified, var cannot be used with the generic type

//       var cannot be used for method parameters and return type
//       var res=method1();

   }
//    static var method1() {return ("Inside Method1");}
//    static method2(var a){System.out.println(a);}

}
```

---

q3

```
package java11;


import java.util.ArrayList;
import java.util.Arrays;


public class J113 {
   public static void main(String[] args) {
      String s="A quick brown fox jumps over the lazy dog";


      String[] strSplit = s.split(" ");


      ArrayList<String> words = new ArrayList<>(Arrays.asList(strSplit));


//       String[] arr=words.toArray(new String[0]);
      String[] arr=words.toArray(String[]::new);


      System.out.println(Arrays.toString(arr));
   }
}
```

Q4

```
package java11;


import java.io.IOException;
```

```
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.List;
import java.util.stream.Collectors;

public class J114 {
   public static void main(String[] args) {
      var path="C:\\Users\\PRATIK\\Desktop\\CG\\StudentList.txt";
      try {
         String data=Files.readString(Path.of(path));

         List<String> s=data.lines().map(t->t.trim()).collect(Collectors.toList());
         s.stream().filter(t-> !t.isBlank()).forEach(t-> System.out.print(t+" "));

         System.out.println();
         long count=s.stream().filter(t-> !t.isBlank()).count();
         System.out.println("Number of students :"+ count);

      } catch (IOException e) {
         e.printStackTrace();
      }

   }
}
```

Q5

```
package java11;


import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.StandardOpenOption;
import java.util.Scanner;



public class J115 {
   public static void main(String[] args) {
      Scanner ip=new Scanner(System.in);
      int total=0, option, price, count=1;

      var path="C:\\Users\\PRATIK\\Desktop\\CG\\Price.txt";
      var path1="C:\\Users\\PRATIK\\Desktop\\CG\\Total.txt";
      String resp="yes";

      do {
         System.out.println("Select your option \n 1: Insert New Price, 2: View Purchase, 3: Exit");
         option=ip.nextInt();

         if(option==1){
```

```
    while(resp.equalsIgnoreCase("yes")){
        System.out.println("Insert price " + count);
        count++;
        price = ip.nextInt();
        total+=price;
        try {
            Files.writeString(Path.of(path),price+ "\n", StandardOpenOption.APPEND);
        } catch (IOException e) {}

        ip.nextLine();
        System.out.println("Do you want to enter more items? (Yes/No)");
        resp=ip.nextLine();
        if(resp.equalsIgnoreCase("no"))
            break;
    }

  }
  if(option==2) {
    System.out.println("Total price of all items is " + total);
    try {
        Files.writeString(Path.of(path1),total+ "\n", StandardOpenOption.APPEND);
    } catch (IOException e) {}

  }
}while(option!=3);
```

```
    }
}
```

Q6

```
package java11;
import java.io.IOException;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class J116 {
    public static void main(String[] args) {
        String uri="https://httpbin.org/get";
        HttpRequest req=HttpRequest.newBuilder()
                .uri(URI.create(uri))
                .GET()
                .version(HttpClient.Version.HTTP_2)
                .build();

        HttpClient client=HttpClient.newBuilder().build();

        try {
           HttpResponse<String> resp= client.send(req, HttpResponse.BodyHandlers.ofString());
            System.out.println("Status code :"+resp.statusCode());
```

```
            System.out.println(resp.body());
            System.out.println(resp.headers());
        } catch (IOException e) {
            e.printStackTrace();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```