

1. What syntax is used to import the entire Java utilities package? And if you import an entire package do you also need to import additional classes in the same package separately? `import java.util.*;` `import pkg.*`
2. Write true or false on the blanks in the program below to show the value of the boolean variable `true_false` as the program executes.

```
int i=5;
int j=6;
boolean true_false;
true_false=(j<5); false
true_false=(j>3); true
true_false=(j<i); false
true_false=(i<5); false
true_false=(j<=5); false
true_false=(6<6); false
true_false=(i!=j); true
true_false=(i==j || i<50); true
true_false=(i==j && i<50); false
true_false=(!(i<2 && j==5)); true
true_false=!true_false; false
```

3. Explain why each of the declarations are wrong
`int 2beOrNot2be;` започва с число
`float price index;` интервал
`double lastYear'sPrice;` апостроф
4. Explain why each of the declarations do not follow conventions for variable names.
`int c=3,s=55,g=4;` не значат нищо
`final double salesTax=0.06;` главни букви и разделено " _"
`double gearratio = 0.5,Gear = 4;` `gearRatio`, `gear`
`int current gear;` интервал
5. Given the three String objects below, what will each of the following return?
`String s1 ="ABC";`
`String s2 = new String("DEF");`
`String s3 = "AB" + "C";`

- a. `s1.compareTo(s2);` -1
- b. `s2.equals(s3);` true
- c. `s3 == s1;` true
- d. `s2.compareTo(s3);` 1

e. s3.equals(s1); true

6. True or False: IF/ELSE statements can always be replaced with SWITCH statements. he

7. Without typing in the code determine the output of the following program.

```
int num[] = {7,7,6,6,5,5,4,4};
for(int i = 0; i < 8; i = i + 2)
    System.out.print(num[i]);
7654
```

8. Without typing in the code determine the output of the following program.

```
int[][] num = {{3,3,3},{2,2,2}};
int[] array = {4,3,2};
for(int i = 0; i < 3; i++){
    num[1][i] = num[0][i]+array[i];
}
for(int i = 0; i < 2; i++){
    for(int j = 0; j < 3; j++){
        System.out.print(num[i][j]);
    }
    System.out.println();
}
333
765
```

9. What is wrong with the following code? It should print "Hello World!" four times to the screen.

```
String str = "Hello World";
for(int i = 0; i < 4; i++){ {
    System.out.println(str);
    str+= "!";
}
str+= "!"; след String str = "Hello World";,
изтриваме ; след for(int i = 0; i < 4; i++)
```

10. Identify the vocabulary word for each definition below

Default	When there is no access modifier. Same access as public, except not visible to other packages.
---------	--

Access modifiers	The keywords used to declare a class, method, or variable as public, private, or protected.
Sub classes	Classes that are more specific subsets of other classes and that inherit methods and fields from more general classes.
Extends	A keyword in Java that allows you to explicitly declare the superclass of the current class.
Encapsulation	A programming philosophy that promotes protecting data and hiding implementation in order to preserve the integrity of data and methods.
Private	Visible only to the class where it is declared
Hierarchy	A structure that categorizes and organizes relationships among ideas, concepts of things with the most general or all-encompassing component at the top and the more specific, or component with the narrowest scope, at the bottom
Public	Visible to all classes
Super classes	Classes that pass down their methods to more specialized classes.
Inheritance	The concept in object-oriented programming that allows classes to gain methods and data by extending another classes fields and methods
Protected	Visible to the package where it is declared and to subclasses in other packages.
UML	A standardized language for modeling systems and structures in programming.
Super	A keyword that allows subclasses to access methods, data, and constructors from their parent class.
Is-a relationship	A helpful term used to conceptualize the relationships among nodes or leaves in an inheritance hierarchy.
Polymorphism	A concept in object oriented programming that allows classes to have many forms and

	behave like their superclasses.
Overriding	Implementing methods in a subclass that have the same prototype (the same parameters, method name, and return type) as another method in the superclass.
Final	A keyword in Java used to limit subclasses from extending a class, overriding methods or changing data.
Final	A keyword in Java used to limit subclasses from extending a class, overriding methods or changing data.
Overloading	Implementing a method with the same name as another method in the same class that has different parameters or a different return type.
Runtime polymorphism	The process by which Java is able to determine which method to invoke when methods have been overridden.
Abstract	A keyword in Java that allows classes to be extended, but the classes cannot be instantiated (constructed) and when applied to methods, dictates that the methods should be implemented in all subclasses of the class.

11. Given the information for the following, determine whether they will result: Always compile, sometimes compile, or does not compile.

```
public interface A
public class B implements A
public abstract class C
public class D extends C
public class E extends B
```

Each class have been initialized, but it is not clear what they have been initialized to:

```
A a = new...
B b = new...
C c = new...
D d = new...
E e = new...
```

The following methods are included:

```
interface A specifies method void methodA()
class C has the abstract method void methodC()
```

Code:	Always Compile, Sometimes Compile, or Does Not Compile?
a = new B();	Always Compile
d = new C();	Does Not Compile
b.methodA();	Always Compile
e.methodA();	Always Compile
c = new C();	Does Not Compile
(D)c.methodC();	Sometimes Compile