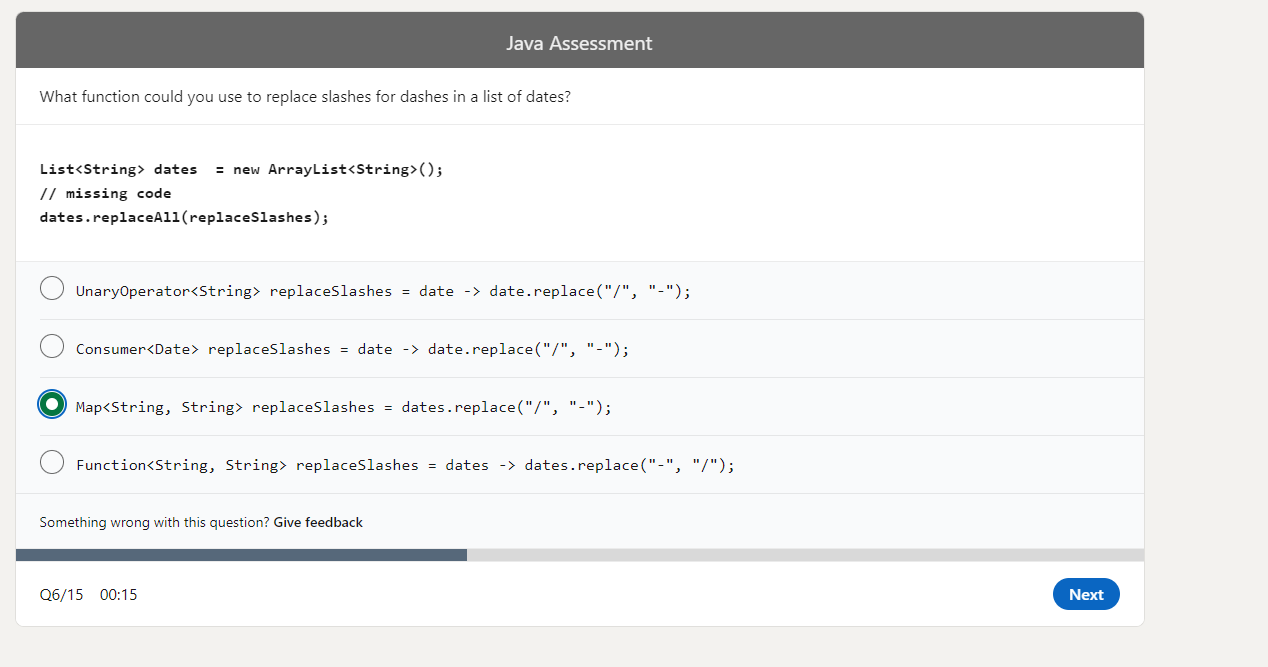
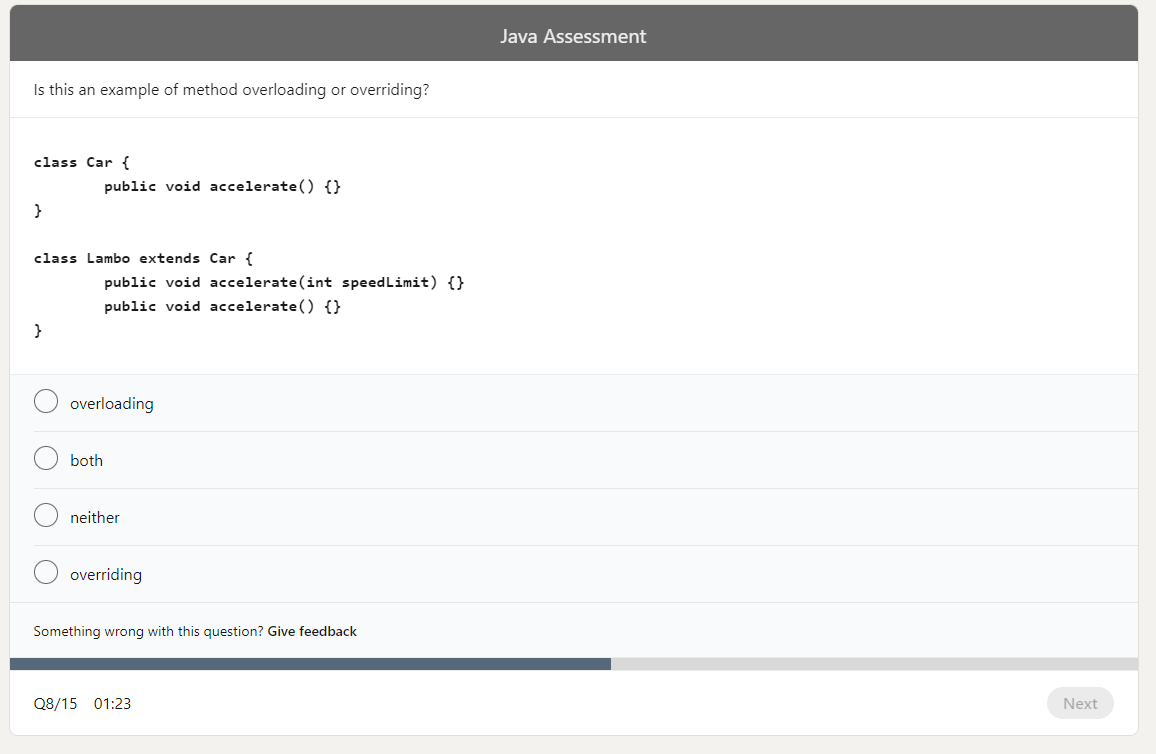
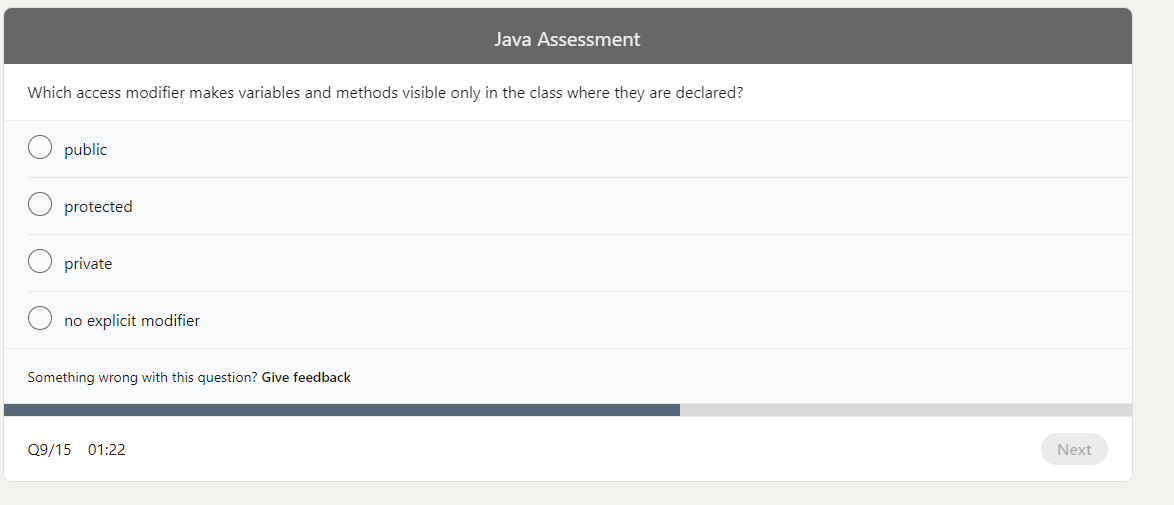
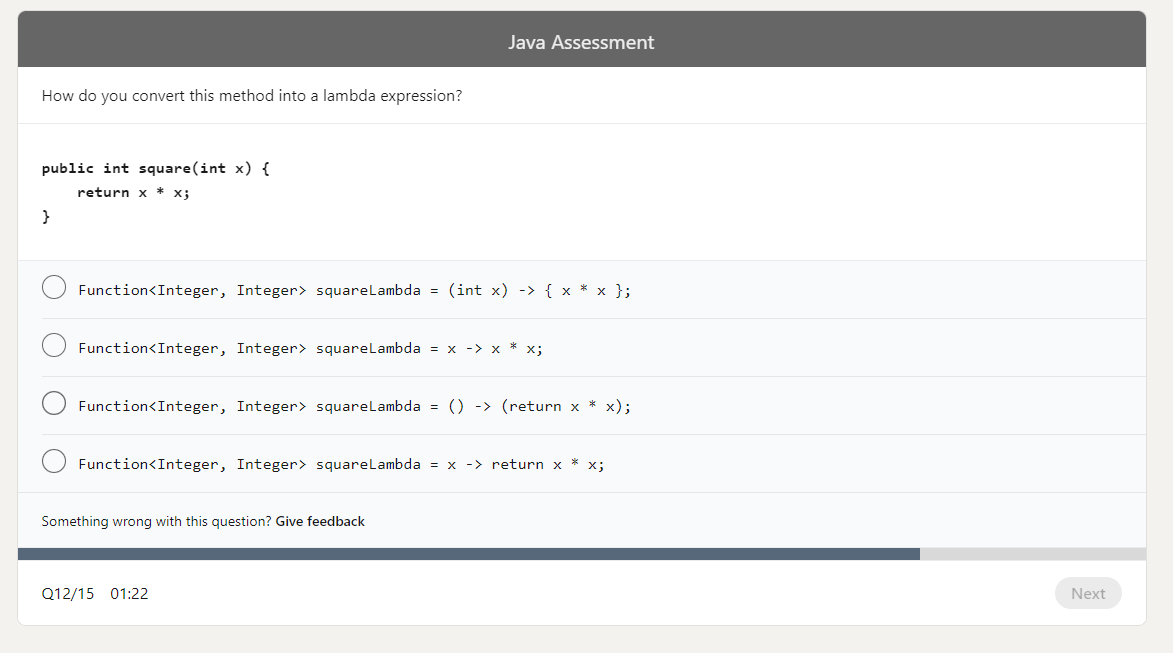
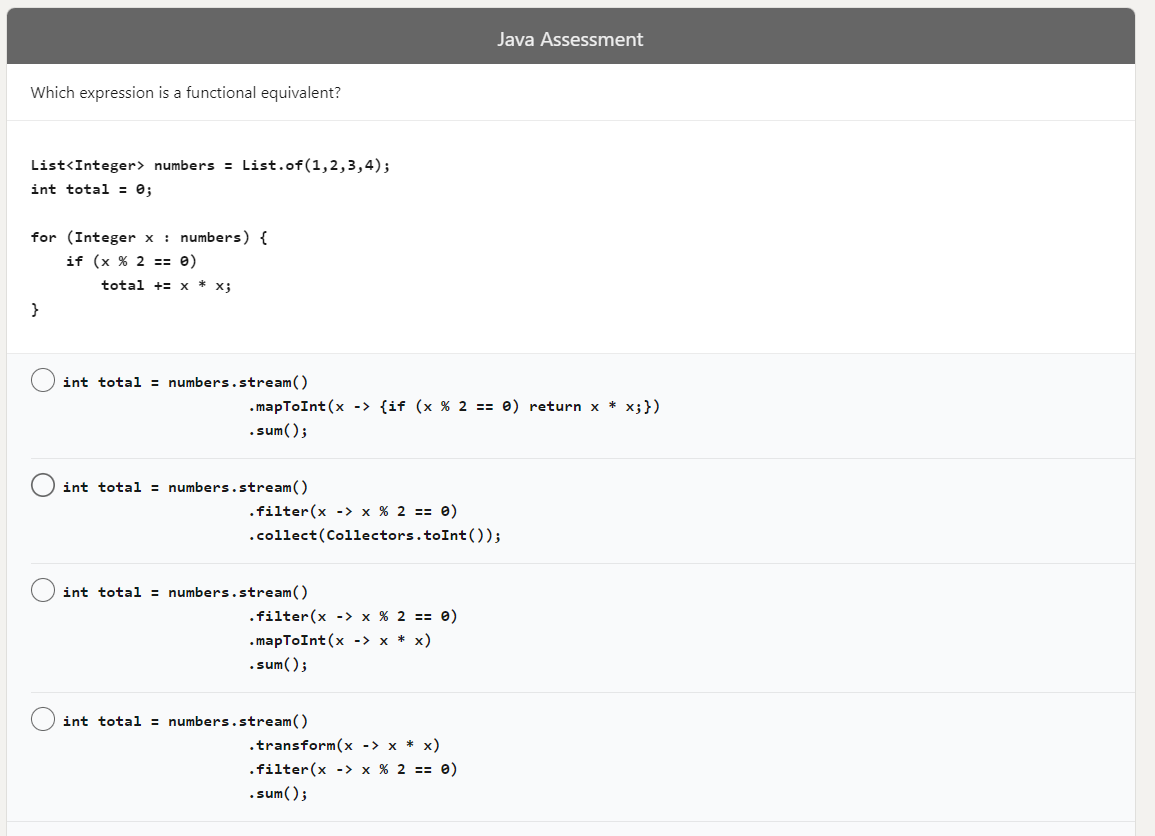
Questions from my very first test – April 2022 (i failed and we don’t know the correct answers)

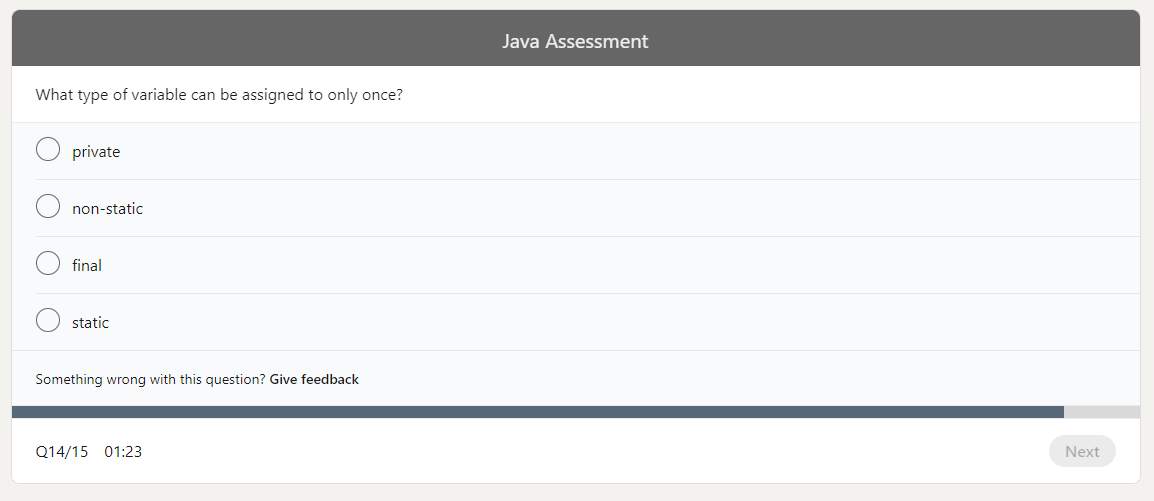


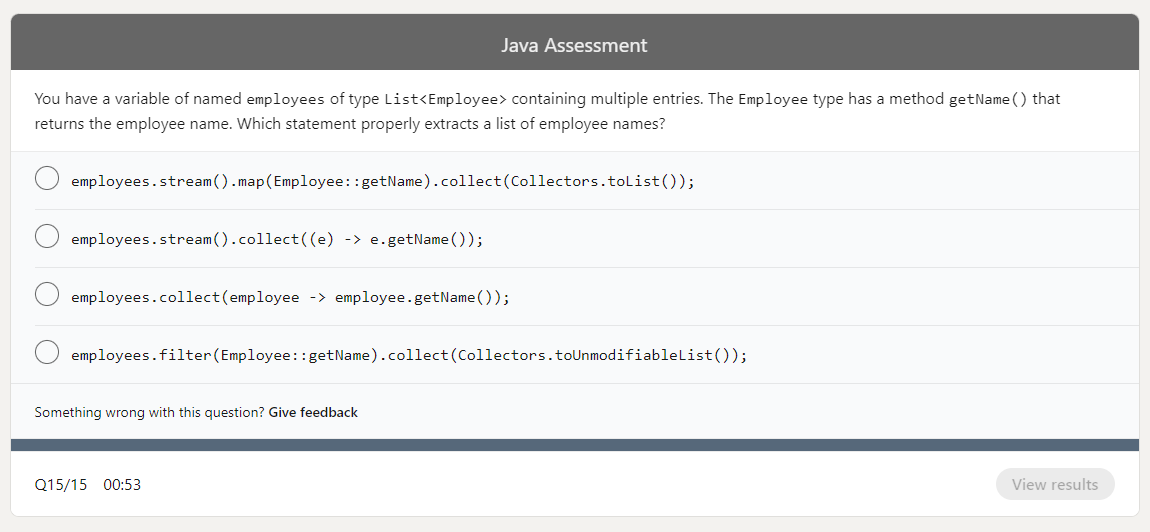












**Questions & Answers of Previous Assesments found on internet:**

https://technorj.com/linkedin-java-skill-assessment-answers-2021/

Q1. What is displayed when this code is compiled and executed?  
Public class main {  
public static void main(string[] args) {  
int x= 5;x = 10;System.out.println(x);  
  }  
}

* 5
* null
* x
* **10 – Correct answers**

**Q2. What statement returns true if “nifty” is of type String?**

* “nifty”.getType().equals(“string”)
* “nifty”.getClass().getSimpleName() == “String”
* “**nifty” instanceof String – Correct Answer**
* “nifty” .getType() == String

**Q3. Given the string “strawberries” saved in a variable called fruit, what would fruit .substring(2, 5) return?**

* **raw – Correct Answer**
* rawb
* traw
* awb

Q4. What is the result of this code?  
try{System.out.print(“Hello World”);}

catch(Exception e) {  System.out.println(“e”);}

catch(ArithmeticException e) { System.out.println(“e”);}

finalyy{ System.out.println(“!”);}

* **It will not compile because the second catch statement is unreachable. – Correct Answer**
* Hello World
* Hellow World!
* It will thwow a runtime exception.

Q5. How many times will this code print “Hello World”?  
class Main{

public static void main(String[] args) {  
 for (int i=0; i<10; i=i++){

i+=1;

System.out.println(“Hello World!”);

}

}

}

* 9 times
* infinite number of times
* 5 times
* **10 times – Correct Answer**

**Q6. What is the result of this code?**  
**class Main {**

**Object message(){ return “Hello!”;}**

**public static void main(String[] args) {**

**System.out.print(new main().MESSAGE());**

**System.out.print(new Main2().message());**

**}**

**}**

**class Main2 extends Main {String message(){ return “World!”; }}**

* It will compile because of line 7.
* Hello!Hello!
* **Hello!World! – Correct Answers**
* It will not compile because of line 11.

**Q7. You have an ArrayList of names that you want to sort alphabetically. Which approach would not work?**

* names.sort(Comparator.comparing(String::toString))
* names = names.stream().sorted((s1, s2) ->

          s1.compareTo(s2)).collect(Collectors.toList())

* **names.sort(List.DECENDING) – Correct Answer**
* Collections.sort(names)

**Q8. What is the output of this code?**  
import java.util.\*;

class Main {

public static void main(String[] args) {

List<Boolean> list = new ArrayList<>();

list.add(true);

list.add(Boolean.parseBoolean(“False”));

list.add(Boolean.TRUE);

System.out.print(list.size());

System.out.print(list.get(1) instanceof Boolean);

}

}

* 3false
* 3true
* 2true
* **A runtime exception is thrown -Correct Answer**

**Q9. What method can be used to create a new instance of an object?**

* another instance
* field
* private method
* **constructor – Correct Answer**

**Q10. How can you achieve runtime polymorphism in Java?**

* method calling
* method overrunning
* method overriding
* **method overloading – Correct Answer**

**Q11. What is the output of this code?**  
class Main {

public static void main(String[] args) {

String message = “Hello wold!”;

String newMessage = message.substring(6, 12) + message.substring(12, 6);

System.out.println(newMessage);

}

}

* The code does not compile
* **A runtime exception is thrown. – Correct Answer**
* world!world!
* world!!world

gv comment: Code compiles ok , but indeed runtime error is thrown during execution because message length is 11 , thus substring(6,12) is invalid.

**Q12. Which is the most reliable expression for testing whether the values of two string variables are the same?**

* string1 == string2
* **string1.equals(string2) – Correct Answer**
* string1 = string2
* string1.matches(string2)

**Q13. What is the output of this code?**  
class Main {  
 static int cound = 0;  
 public static void main(String[] args)  {  
if(cound <3)  
{count++;main(null);}  
else{

return;  
}

System.out.println(“Hello World”);  
 }  
}

* it will throw a runtime exception.
* it will print “Hello World!” three times.
* it will not compile.
* **it will run forever. – Correct Answer**

gv comment: If the code is not wrongly copy-pasted and count++ is indeed written in the real assesment question, then this code will not compile since count is not declared.

If count++ is a typo and should be cound++ then it will just print Hello World tree times.

**Q14. What is the output of this code?**  
**class main {**  
 **public static void main(String[] args) {**

List list = new ArrayList();

**list.add(“hello”);**  
 **list.add(2);**  
 **System.out.print(list.get(0) instanceof Object);**

**System.out.print(list.get(1) instanceof Integer);**  
**}**  
**}**

* falsetrue
* The code does not compile.
* **truetrue – Correct Answer** (gv confirmed . Ps: add import java.util.\*;)
* truefalse

**Q15. By implementing encapsulation, you cannot directly access the class’s\_\_\_\_\_properties unless you are writing code inside the class itself.**

* **private – Correct Answer**
* protected
* public
* no-modifier

### Linkedin Java Assessment Questions and Answers Old

**Q1. Given the string “strawberries” saved in a variable called fruit, what would “fruit.substring(2, 5)” return?**

* rawb
* **raw <<<<—Correct**
* awb
* traw

**Q2. How can you achieve runtime polymorphism in Jav**a?

* method overloading
* method overrunning
* **method overriding <<<<— Correct**
* method calling

**Q3. Given the following definitions, which of these expressions will NOT evaluate to true?**  
**boolean b1 = true, b2 = false;**  
**int i1 = 1, i2 = 2;**

* (i1 | i2) == 3
* **i2 && b1 <<<<—Correct**
* b1 || !b2
* (i1 ^ i2) < 4

**Q4. What can you use to create new instances in Java?**

* **constructor <<<<—Correct**
* another instance
* field
* private method

**Q5. What is the output of this code?**  
**class Main {  
 public static void main (String[] args) {  
     int array[] = {1, 2, 3, 4};  
     for (int i = 0; i < array.size(); i++) {  
        System.out.print(array[i]);  
       }  
     }  
   }**

* **It will not compile because of line 4. <<<<—Correct**
* It will not compile because of line 3.
* 123
* 1234

**Q6. Which of the following can replace the CODE SNIPPET to make the code below print “Hello World”?**  
**interface Interface2 {  
    static void print() {  
        System.out.print(“World!”);  
    }  
}**

* super1.print(); super2.print();
* this.print();
* super.print();
* Interface1.print(); Interface2.print();

**Q7. What does the following code print?String str = “”abcde””;str.trim();str.toUpperCase();str.substring(3, 4);System.out.println(str);**

* CD
* CDE
* D
* “**abcde” <<<<—Correct**

**Q8. What is the result of this code?**  
**class Main {**  
   **public static void main (String[] args){**  
     **System.out.println(print(1));**  
   **}**  
   **static Exception print(int i){**  
       **if (i>0) {**  
          **return new Exception();**  
       **}**  
**else {**  
          **throw new RuntimeException();**  
      **}**  
  **}**  
**}**

* It will show a stack trace with a runtime exception.
* “**java.lang.Exception” <<<<—Correct**
* It will run and throw an exception.
* It will not compile.

**Q9. Which class can compile given these declarations?**  
**interface One {  
      default void method() {  
        System.out.println(“”One””);  
      }   }  
interface Two {  
      default void method () {  
        System.out.println(“”One””);  
    }  
   }**

* class Three implements One, Two {

    publc void method() {        super.One.method();  } }

* class Three implements One, Two {

    publc void method() {        One.method();  } }

* class Three implements One, Two {

}

* **class Three implements One, Two { <—— correct**

**publc void method() {**        **One.super.method();**  **} }**

**Q10. What is the output of this code?**  
**class Main {  
  public static void main (String[] args) {  
      List list = new ArrayList();  
      list.add(“hello”);  
      list.add(2);  
    System.out.print(list.get(0) instanceof Object);  
      System.out.print(list.get(1) instanceof Integer);  
  }  
}**

* The code does not compile.
* truefalse
* **truetrue <<<<—Correct**
* falsetrue

**Q11. Given the following two classes, what will be the output of the Main class?**  
**package mypackage;**  
**public class Math {**  
    **public static int abs(int num){**  
        **return num<0?-num:num;**  
    **}**  
**}**  
**package mypackage.elementary;**  
**public class Math {**  
    **public static int abs (int num) {**  
        **return -num;**  
    **}**  
**}**  
**import mypackage.Math;**  
**import mypackage.elementary.\*;  
class Main {**  
  **public static void main (String args[]){**  
    **System.out.println(Math.abs(123));**  
  **}**  
**}**

* Lines 1 and 2 generate compiler erros due to class name conflicts.
* “-123”
* It will throw an exception on line 5.
* “**123” <— Correct // The answer is “123”. The abs() method evaluates to the one inside mypackage.Math class.**

**Q12. What is the result of this code?**  
**class MainClass {  
  final String message(){  
      return “Hello!”;  
  }  
}  
class Main extends MainClass {  
  public static void main(String[] args) {  
      System.out.println(message());  
  }  
String message(){  
     return “World!”;  
  }  
}**

* **It will not compile because of line 10. <— Correct**
* “Hello!”
* It will not compile because of line 2.
* “World!”

**Q13. Given this code, which command will output “2”?**  
**class Main {**  
    **public static void main(String[] args) {**  
        **System.out.println(args[2]);**  
    **}**  
**}**

* java Main 1 2 “3 4” 5
* **java Main 1 “2” “2” 5 <— Correct**
* java Main.class 1 “2” 2 5
* java Main 1 “2” “3 4” 5

**Q14. What is the output of this code?**  
**class Main {    public static void main(String[] args){        int a = 123451234512345;        System.out.println(a);    }}**

* “123451234512345”
* **Nothing – this will not compile. <<<<—Correct**
* a negative integer value
* “12345100000”

**Q15. What is the output of this code?**

**class Main {    public static void main (String[] args) {        String message = “Hello world!”;        String newMessage = message.substring(6, 12)            + message.substring(12, 6);        System.out.println(newMessage);    }}**

* The code does not compile.
* **A runtime exception is thrown <<<<—Correct**
* “world!!world”
* “world!world!”
* String m = “Hello world!”;
* String n = m.substring(6,12) + m.substring(12,6);
* System.out.println(n);

**Q16. How do you write a foreach loop that will iterate over ArrayList<Pencil>pencilCase?**

* for(Pencil pencil = pencilCase){}
* Iterator iterator = pencilCase.iterator();
* for(){iterator.hasNext()}{}

**Q17. Fill in the blanks?**

**Object-oriented programming (OOP) is a programming language model that organizes software design around (objects), rather than (actions).**

**Q18. What code would you use to tell if “schwifty” is of type String?**

* “schwifty”.getType() == String
* “schwifty”.getClass().getsimpleName() == “String”
* “schwifty”.getType().equals(“String”)
* “**schwifty” instanceof String <<<<—Correct**

**Q19. Correct output of “apple”.compareTo(“banana”)**

* 0
* positive number
* **negative number <<<<—Correct**
* compilation error

**Q20. You have an ArrayList of names that you want to sort alphabetically. Which approach would NOT work?**

* names.sort(Comparator.comparing(String::toString))
* Collections.sort(names)
* **names.sort(List.DESCENDING) <<<— Correct (not too sure)**
* names.stream().sorted((s1, s2) -> s1.compareTo(s2)).collect(Collectors.toList())

**Q21. By implementing encapsulation, you cannot directly access the class’s \_\_\_\_\_ properties unless you are writing code inside the class itself.**

* **private <<<<—Correct**
* protected
* no-modifier
* public

**Q22. Which is the most up-to-date way to instantiate the current date?**

* new SimpleDateFormat(“yyyy-MM-dd”).format(new Date())
* new Date(System.currentTimeMillis())
* LocalDate.now()
* Calender.getInstance().getTime() <<<<— Correct

**Q23. Fill in the blank to create a piece of code that will tell wether int0 is divisible by 5:**

* boolean isDivisibleBy5 = \_\_\_\_\_
* int0 / 5 ? true: false
* **int0 % 5 == 0 <<<<—Correct**
* int0 % 5 != 5
* Math.isDivisible(int0, 5)

**Q24. How many time will this code print “Hello World!”?**  
**Class Main {  
    public static void main(String[] args){  
        for (int i=0; i<10; i=i++){   
          i+=1;  
          System.out.println(“Hello World!”);  
        }  
    }  
}**

* 10 times
* 9 times
* **5 times <<<<—Correct**
* infinite number of times

GV Comment : This code when compiled printed Hello World 10 times, not 5 times.

**Q25. The runtime system starts your program by calling which function first?**

* print
* iterative
* hello
* **main <<<<—Correct**

**Q26. What is the result of this code?**  
**try{  
    System.out.print(“Hello World”);  
}catch(Exception e){  
    System.out.println(“e”);  
}catch(ArithmeticException e){  
    System.out.println(“e”);  
}finally{  
    System.out.println(“!”);  
}**

* It will throw a runtime exception
* **It will not compile <<<<—Correct**
* Hello World!
* Hello World

**Q27. Which statement is NOT true?**

* An anonymous class may specify an abstract base class as its base type.
* **An anonymous class does not require a zero-argument constructor. <<<<—Correct**
* An anonymous class may specify an interface as its base type.
* An anonymous class may specify both an abstract class and interface as base types

**Q28. What will this program print out to the console when executed?**  
**public class Main {**  
**public static void main(String[] args){**  
       **LinkedList<Integer> list = new LinkedList<>();**  
       **list.add(5);**  
       **list.add(1);**  
       **list.add(10);**  
       **System.out.println(list);**  
    **}**  
**}**

* **[5, 1, 10] <<<<—Correct**
* [10, 5, 1]
* [1, 5, 10]
* [10, 1, 5]

**Q29. What is the output of this code?**  
**class Main {**  
**public static void main(String[] args){**  
        **String message = “Hello”;**  
**for (int i = 0; i<message.length(); i++){**  
**System.out.print(message.charAt(i+1));**  
**}**  
**}**  
**}**

* “Hello”
* **A runtime exception is thrown. <<<<—Correct**
* The code does not compile.
* “ello”

**Q30. Object-oriented programming is a style of programming where you organize your program around \_\_\_\_ rather than \_\_\_\_ and data rather than logic.**

* functions; actions
* objects; actions
* actions; functions
* actions; objects

Sources:

https://www.compilejava.net/

<https://www.linkedin.com/learning/search?keywords=Java>

<https://stackoverflow.com/tags/java/info>

<https://www.linkedin.com/feed/update/urn:li:lyndaCourse:2825378/>

<https://www.linkedin.com/feed/update/urn:li:lyndaCourse:375490/>

<https://www.linkedin.com/feed/update/urn:li:lyndaCourse:2453107/>

<https://www.linkedin.com/feed/update/urn:li:lyndaCourse:808674/>

<https://stackoverflow.com/questions/2857376/difference-between-java-se-ee-me>

<https://hackr.io/blog/java-certification-courses>

sudo update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk-18/bin/java 1112

this works ok and in reality what it does under the hood is to update the /*usr*/bin/java link

Then run:

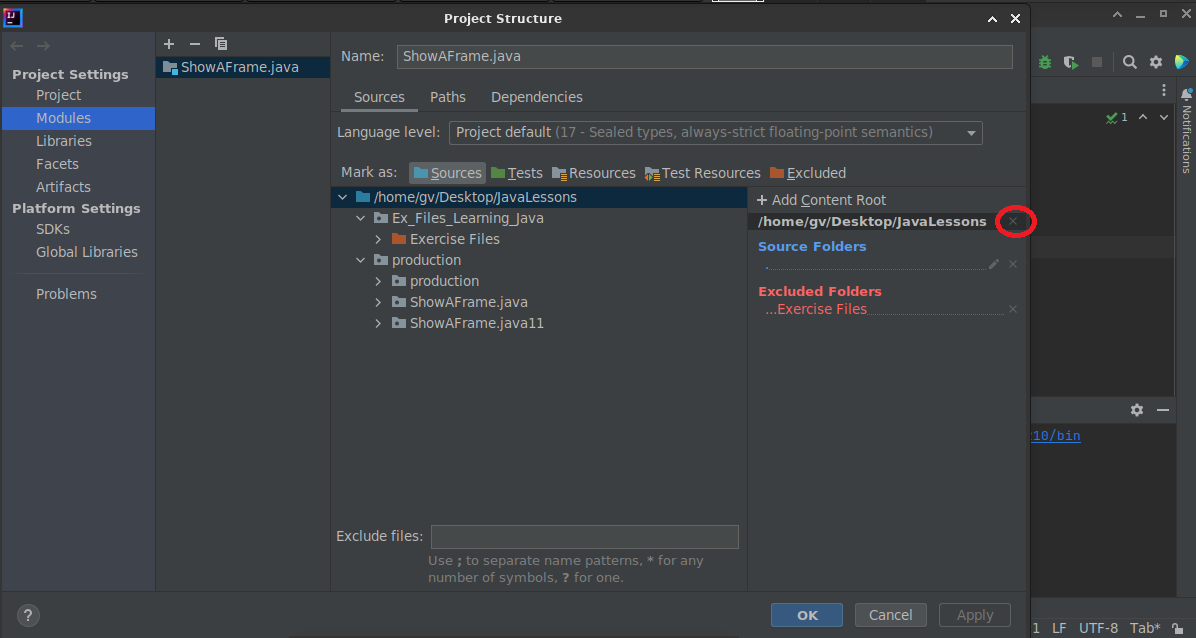
sudo update-alternatives --config java and select the new added entry.

Confirm by just running java --version and you should see the newly selected entry.

Configure IntelliJ to run a java file from anywhere

IntelliJ (and also eclipse) works with “Projects”. When you define a project, the relevant project directories are created and used (i.e src for source, etc).

When a file out of the project is opened , IntelliJ does not recognize it and can not run it or correct synthax error. The workaround is to add the directory of your file to the project roots. To do so go to “Project Settings” gear on the top right of your IntelliJ , open it , Project Strutcture, go to Modules , close as shown bellow previous root and then select add root and select the folder that your standalone java file resides.



Object Class: i did create the Triangle Object Class using the LinkedIn Tutoral , by right click on Project Name, new class file -> Triangle -> Triangle.java file was created by IntelliJ.

On the next compile, Triangle.class was created in the project “out” directory, together with the rest classes of the same project. If you delete the .class file from there, then it seems that there is no way to make the project work ok as before.

Workaround: Use terminal, goto to Project home directory where Triangle.java is located, compile it (i.e javac Triangle.java) , the Triangle.class is created, and manually copy the class file into the Project out folder.

$ cp Triangle.class ./production/ShowAFrame.java/out/production/ShowAFrame.java/Triangle.class

You can find the correct path from IntelliJ - Project settings - Project Structure - Project Tab - Compiler Output

Tip: if you manually put the main java file /class file and the Triangle.java and .class into a new tmp directory , you can run normally the main class (without errors for not finding Triangle) using geany - compile - run command buttons.

**Java Method Overload:** Let’s you define a second method / function with the same name but different parameters, or diferent return type.

**Functions Return Type:** When you add methods in class /objects , you have either to declare the return type of the method (i.e public double FindTriangleArea(...)) or if you want the method to be stand alone you have to declare it as void :

public void SetNewLength(double NewLength) {this.length=NewLength;};

**Java Method Override:** When working with sub-classes, you can have a method in the sub-class with same method name that exists in parent class but with different functionality.

Tip: You can combine Override + Overflow in the same sub-class

Benefits of Method Override: Parent Class Bank , method TaxInterest , sub-class AXXA , override TaxInterest inside AXXA sub-class and assign a different rate.

Example of Method Override and Overload:

class A {

public void doSth() { /// }

}

class B extends A {

public void doSth() { /\* method overriden \*/ }

public void doSth(String b) { /\* method overloaded \*/ }

}

[gv]: Why to create sub-classes and override methods while it is possible to declare AXXA and other banks as Bank AXXA = new Bank(interest);

System.out.println(AXXA.TaxInterest); will refer to axxa object , while nbg.taxinterest will refer to taxinterest of nbg class, etc.

***Java Classes vs Interfaces***

***Extends keyword***: This is used to get attributes of a parent class into child class and may contain already defined methods that can be overridden in the child class.

***Implements keyword***: This is used to implement an interface (parent class with functions signatures only but not their definitions) by defining it in the child class.

Java Access Modifiers for variables, methods, etc

https://www.tutorialspoint.com/java/java\_access\_modifiers.htm

\* No Keyword: Available in all class of the same project

\* Private: Variables accessible only by the class that creates them. if public getter methods are present in the class, then those private variables can be also exposed to the other classes.

\* public : A class, method, constructor, interface, etc. declared public can be accessed from any other class, even not in the same package (but requires importing)

\* protected

\*

**Sub-Classes**

A subclass is a class that extends another class.Inheritance is a way for classes to add specialized behavior ontop of generalized behavior. This is often represented by the phrase "is a" relationship.

Java Keywords:

this : refers to the current class / current object.