Index

Java

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA),[meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems(which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its original features from Small Talk, with a syntax similar to C and C++, but it has fewer low-level facilities than either of them.

Advantages:

❖ Java is easy to learn.

❖ Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.

❖ Java is object-oriented. his allows you to create modular programs and reusable code.

❖Java is pltform-independent. One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.

Program Components

The three main components of java

languages are:

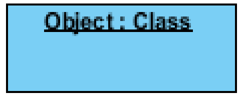
JVM-Java Virtual Machine.

JDK-Java Development Kit.

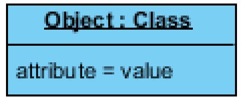
JRE-Java Runtime Environment .

Object Diagram Symbols and Notations

Object Names:
•Every object is actually symbolized like a rectangle, that offers the name from the object and its class underlined as well as divided with a colon.

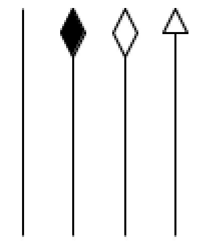


Object Attributes:
•Similar to classes, you are able to list object attributes inside a separate compartment. However, unlike classes, object attributes should have values assigned for them.

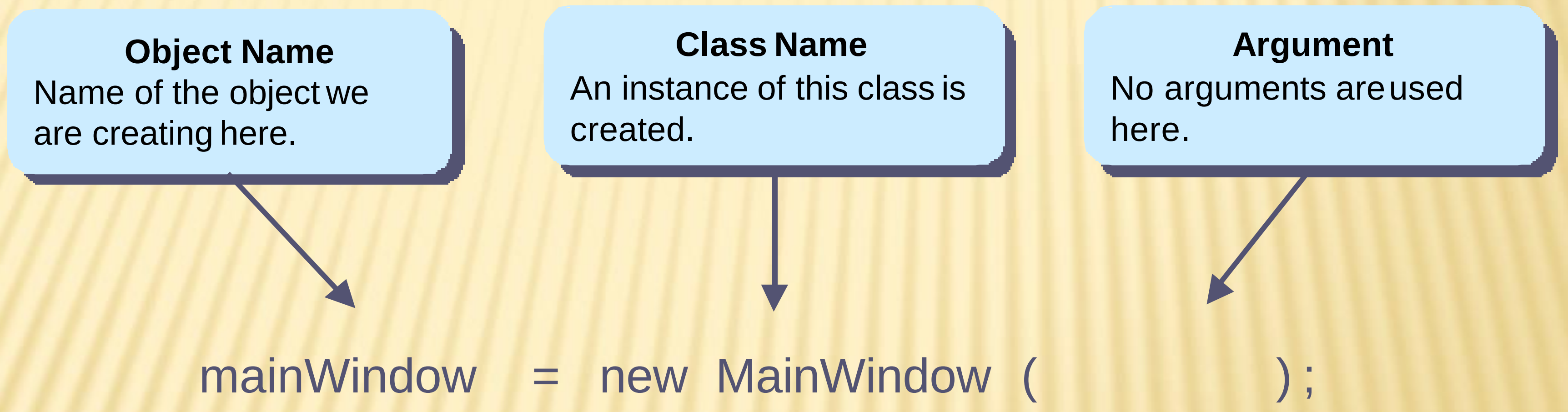


Links:

•Links tend to be instances associated with associations. You can draw a link while using the lines utilized in class diagrams.



Object Creation



Object Name
Name of the object we are creating here.

Class Name
An instance of this class is created.

Argument
No arguments are used here.

```
mainWindow   =  new  MainWindow  (              ) ;
```

Three types of comments

```
/*
    This is a comment with
    three lines of
    text.
*/
```

Multiline Comment

```
// This is a comment
// This is another comment
// This is a third comment
```

Single line Comments

```
/**
 * This class provides basic clock functions. In addition
 * to reading the current time and today's date, you can
 * use this class for stopwatch functions.
 */
```

javadoc Comments

Built-In Types Of Variables

Type	Description
byte	8 bit signed integer
short	16 but signed integer
int	32 bit signed integer
long	64 bit signed integer
float	32 bit signed real number
double	64 bit signed real number
char	16 bit Unicode character (ASCII and beyond)
boolean	1 bit true or false value
String	A sequence of characters between double quotes ("")

Keywords in Java

❖ **Abstracts** are used to implement an abstraction in Java. A method with no definition must be declared as abstract and the class containing it must be declared as abstract. Abstract classes cannot be instantiated. Abstract methods must be implemented in the sub classes. The abstract keyword cannot be used with variables or constructors. Note that an abstract class isn't required to have an abstract method at all.

❖ **Assert** describes a predicate (a true–false statement) placed in a Java program to indicate that the developer thinks that the predicate is always true at that place. If an assertion evaluates to false at run-time, an assertion failure results, which typically causes execution to abort. Optionally enable by Class Loader method Boolean.

❖ **Boolean** defines a Boolean variable for the values "true" or "false" only. By default, the value of boolean primitive type is false. This keyword is also used to declare that a method returns a value of the primitive type Boolean.

❖ **Break** Used to end the execution in the current loop body.

❖ The **byte** keyword is used to declare a field that can hold an 8-bit signed two's complement integer. This keyword is also used to declare that a method returns a value of the primitive type byte.

❖ A statement in the switch block can be labeled with one or
more case or default labels. The switch statement evaluates its expression, then executes all statements that follow the matching **case** label; see switch.

❖ **Catch** Used in conjunction with a try block and an optional finally block. The statements in the catch block specify what to do if a specific type of exception is thrown by the try block.

❖ **Character** defines a character variable capable of holding any character of the java source file's character set.

❖ **Continue** Used to resume program execution at the end of the current loop body. If followed by a label, continue resumes execution at the end of the enclosing labeled loop body.

- **Default** keyword can optionally be used in a switch statement to label a block of statements to be executed if no case matches the specified value; see *switch. Alternatively*, the default keyword can also be used to declare default values in a Java annotation. From Java 8 onwards, the default keyword is also used to specify that a method in an interface provides the default implementation of a method.

- A **type** that defines the implementation of a particular kind of object. A class definition defines instance and class fields, methods, and inner classes as well as specifying the interfaces the class implements and the immediate super class of the class. If the super class is not explicitly specified, the super class is implicitly Object. The class keyword can also be used in the form Class**.class** to get a Class object without needing an instance of that class. For example, **String. class** can be used instead of doing **new String().get Class()**..

- The **double** keyword is used to declare a variable that can hold a 64-bit double precision IEEE 754 looting-point number. This keyword is also used to declare that a method returns a value of the primitive type double.[

- The **do keyword** is used in conjunction with while to create a do-while loop, which executes a block of statements associated with the loop and then tests a Boolean expression associated with the while. If the expression evaluates to true, the block is executed again; this continues until the expression evaluates to false.

- The **else keyword** is used in conjunction with if to create an if-else statement, which tests a Boolean expression, if the expression evaluates to true, the block of statements associated with the if are evaluated; if it evaluates to false, the block of statements associated with the else are evaluated.

- The **long keyword** is used to declare a variable that can hold a 64-bit signed two's complement integer. This keyword is also used to declare that a method returns a value of the primitive type long.[

- The **module keyword** is used to declare a module inside of a Java application. This keyword is only available in Java 9 and later.

- **native** Used in method declarations to specify that the method is not implemented in the same Java source file, but rather in another language.

Java **package** is a group of similar classes and interfaces. Packages are declared with the package keyword.

The **private** keyword is used in the declaration of a method, field, or inner class; private members can only be accessed by other members of their own class.[

The **protected** keyword is used in the declaration of a method, field, or inner class; protected members can only be accessed by members of their own class, that class's subclasses or classes from the same package.

] The **public** keyword is used in the declaration of a class, method, or field; public classes, methods, and fields can be accessed by the members of any class.[

Requires Used to specify the required libraries inside of a module.[14] This keyword is only available in Java 9 and later.

Return Used to finish the execution of a method. It can be followed by a value required by the method definition that is returned to the caller.

Project - Introduction

This project is proposed to students in order to evaluate their skills for the fundamental period of Java Programming and UML.

This goal is achieved through the realization of a console application (+ GUI as possible bonus), which aims at managing digital quiz preparation and execution.

Project - Specifications

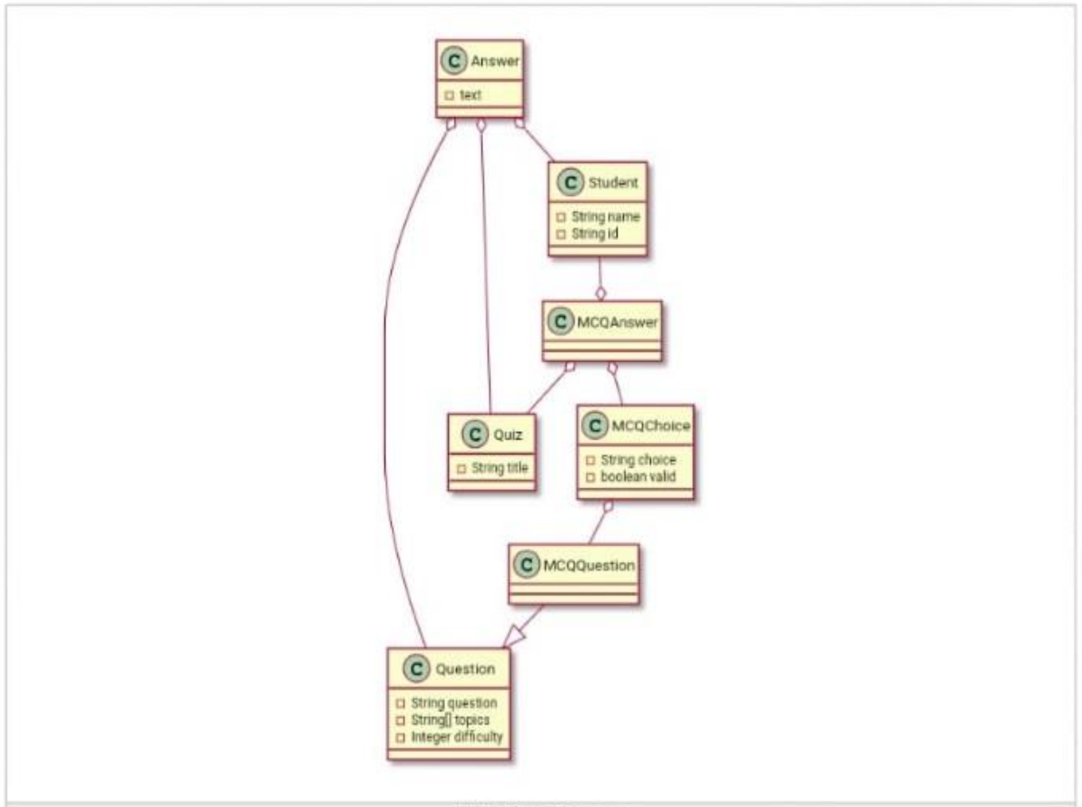
The usual problem while preparing and running an evaluation, is to :

Constitute an appropriate evaluation corresponding of the required level

Reuse former questions

Organize sample evaluations

Correct automatically the MCQ questions.



UML Class Diagram

To handle the most of the possible cases, there are several types of question to consider.
•MCQ Questions
•Open Questions
•Associative Questions

MCQ questions:

The MCQ questions are composed of a question text and a set of possible choices, each choice can be right or wrong. It can also be interesting to add a extra content, like some code extract, some picture or some other kind of media (video, music etc.).

Open Questions:

The open questions are composed only by a question, and some hints, additionally they can be completed by a extra media content.

Associative questions:

The associative questions are questions where it necessary to assign some propositions to some descriptions, like in the following.

Common questions attributes:

Each question has a some extra attributes to describe the **topic** (tag) and the **difficulty** of the question. Those two fields help to balance the overall exam complexity, and the topics coverage. Those attributes can be taken in account for automatic exam assembly.

Project (Questions xml) - Code

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<questions>
        <question order="1">
                <id>1</id>
    <label>Decrement Operator, −−, Decreases The Value
Of Variable By What Number?</label>
    <difficulty>1</difficulty>
    <topics>
      <topic>Java</topic>
    </topics>
    <choices>
      <choice valid="false"> 1 </choice>
      <choice valid="false"> 2 </choice>
      <choice valid="false"> 3 </choice>
      <choice valid="true"> 4 </choice>
    </choices>
        </question>
```

```
<question order="2">
                    <id>2</id>
                    <label>Which Of The Following Can Be
Operands Of Arithmetic Operators?</label>
                    <difficulty>3</difficulty>
                    <topics>
                              <topic>Java</topic>
                    </topics>
                    <choices>
                              <choice valid="false"> Numeric
</choice>
                              <choice valid="false"> Boolean
</choice>
       <choice valid="false"> Characters </choice>
       <choice valid="true"> Both Numeric and Characters
</choice>
                    </choices>
          </question>
```

```
<question order="3">
				<id>3</id>
	<label>Which Of These Is Supported By Method
Overriding In Java?</label>
				<difficulty>1</difficulty>
				<topics>
						<topic>Java</topic>
				</topics>
				<choices>
						<choice valid="true">
Abstraction </choice>
						<choice valid="false">
Encapsulation </choice>
						<choice valid="false">
Polymorphism </choice>
		<choice valid="false"> None of the Mentioned
</choice>
				</choices>
			</question>
```

```
<question order="4">
				<id>4</id>
				<label>Which Of This Keyword Can Be
Used In A Subclass To Call The Constructor Of
Superclass?</label>
				<difficulty>3</difficulty>
				<topics>
					<topic>Java</topic>
				</topics>
				<choices>
					<choice valid="true"> super
</choice>
					<choice valid="flase"> this
</choice>
					<choice valid="false"> extends
</choice>
					<choice valid="false"> extent
</choice>
				</choices>
		</question>
```

```
<question order="5">
                    <id>5</id>
                    <label>Which Data Type Value Is
Returned By All Transcendental Math Functions?</label>
                    <difficulty>3</difficulty>
                    <topics>
                              <topic>Java</topic>
                    </topics>
                    <choices>
                              <choice valid="false"> int
</choice>
                              <choice valid="false"> float
</choice>
                              <choice valid="true"> double
</choice>
                              <choice valid="false"> long
</choice>
                    </choices>
          </question>
```

```
<question order="6">
		<id>6</id>
		<label>What Is The Range Of Short Data Type In Java?</label>
		<difficulty>3</difficulty>
		<topics>
			<topic>Java</topic>
		</topics>
		<choices>
			<choice valid="false"> -128 To 127 </choice>
			<choice valid="true"> -32768 To 32767 </choice>
			<choice valid="false"> -2147483648 To 2147483647 </choice>
			<choice valid="false"> None Of The Mentioned </choice>
		</choices>
	</question>
```

```xml
 <question order="7">
        <id>7</id>
        <label>JVM Stands For? (Type Your Answer)</label>
        <difficulty>2</difficulty>
        <topics>
            <topic>Java</topic>
        </topics>
        <answers>
            <answer>Java Virtual Machine</answer>
        </answers>
    </question>
<question order="8">
        <id>8</id>
        <label>Can 8 byte long data type be automatically type
cast to 4 byte float data type?</label>
        <difficulty>3</difficulty>
        <topics>
            <topic>Java</topic>
        </topics>
        <choices>
            <choice valid="true"> True </choice>
            <choice valid="flase"> Flase </choice>
        </choices>
    </question>
```

```
<question order="9">
        <id>9</id>
        <label>What Is True About Class.getinstance()?</label>
        <difficulty>3</difficulty>
        <topics>
            <topic>Java</topic>
        </topics>
        <choices>
            <choice valid="false"> Class.getinstance Calls The
Constructor </choice>
            <choice valid="flase"> Class.getinstance Is Same As
New Operator </choice>
            <choice valid="flase"> Class.getinstance Needs To
Have Matching Constructor </choice>
            <choice valid="true"> Class.getinstance Creates
Object If Class Does Not Have Any Constructor </choice>
        </choices>
    </question>
```

```
<question order="10">
        <id>10</id>
        <label>What Is True About Constructor?</label>
        <difficulty>3</difficulty>
        <topics>
            <topic>Java</topic>
        </topics>
        <choices>
            <choice valid="true"> It Can Contain Return Type
</choice>
            <choice valid="flase"> It Can Take Any Number Of
Parameters </choice>
            <choice valid="flase"> It Can Have Any Non Access
Modifiers </choice>
            <choice valid="flase"> Constructor Cannot Throw An
Exception </choice>
        </choices>
    </question>
```

```
<question order="11">
        <id>11</id>
        <label>PHP Stands For? (Type Your Answer)</label>
        <difficulty>2</difficulty>
        <topics>
            <topic>PHP</topic>
        </topics>
        <answers>
            <answer>Personal Home Page</answer>
        </answers>
    </question>
    <question order="12">
        <id>12</id>
        <label>Which Version Of PHP Introduced Try/catch
Exception?</label>
        <difficulty>3</difficulty>
        <topics>
            <topic>PHP</topic>
        </topics>
        <choices>
            <choice valid="false"> PHP 4 </choice>
            <choice valid="true"> PHP 5 </choice>
            <choice valid="flase"> PHP 5.3 </choice>
            <choice valid="flase"> PHP 6 </choice>
        </choices>
```

```
<question order="13">
        <id>13</id>
        <label>Type Hinting Was Introduced In Which Version Of
PHP?</label>
        <difficulty>3</difficulty>
        <topics>
            <topic>PHP</topic>
        </topics>
        <choices>
            <choice valid="false"> PHP 4 </choice>
            <choice valid="true"> PHP 5 </choice>
            <choice valid="flase"> PHP 5.3 </choice>
            <choice valid="flase"> PHP 6 </choice>
        </choices>
    </question>
```

```xml
<question order="14">
        <id>14</id>
        <label>How Many Error Levels Are Available In
PHP?</label>
        <difficulty>3</difficulty>
        <topics>
            <topic>PHP</topic>
        </topics>
        <choices>
            <choice valid="false"> 14 </choice>
            <choice valid="false"> 15 </choice>
            <choice valid="true"> 16 </choice>
            <choice valid="flase"> 17 </choice>
        </choices>
    </question>
    <question order="15">
        <id>15</id>
        <label>Which One Of The Following Is Displayed Below
The Class Name In The Class Diagrams?</label>
        <difficulty>3</difficulty>
        <topics>
            <topic>PHP</topic>
            <topic>UML Diagram</topic>
        </topics>
        <choices>
```

```
 <choice valid="false"> Functions </choice>
           <choice valid="false"> Methods </choice>
           <choice valid="true"> Attributes </choice>
           <choice valid="flase"> Constraints </choice>
       </choices>
   </question>
<question order="16">
       <id>16</id>
       <label>Kind Of Diagrams Which Are Used To Show
Interactions Between Series Of Messages Are Classified
As</label>
       <difficulty>3</difficulty>
       <topics>
           <topic>UML Diagram</topic>
       </topics>
       <choices>
           <choice valid="false"> Activity Diagrams </choice>
           <choice valid="false"> State Chart Diagrams
</choice>
           <choice valid="true"> Collaboration Diagrams
</choice>
           <choice valid="flase"> Object Lifeline Diagrams
</choice>
       </choices>
   </question>
```

```
<question order="17">
        <id>17</id>
        <label>Diagrams Which Are Used To Distribute Files,
Libraries And Tables Across Topology Of Hardware Are
Called</label>
        <difficulty>3</difficulty>
        <topics>
            <topic>UML Diagram</topic>
        </topics>
        <choices>
            <choice valid="true"> Deployment Diagrams
</choice>
            <choice valid="false"> Use Case Diagrams </choice>
            <choice valid="false"> Sequence Diagrams </choice>
            <choice valid="flase"> Collaboration Diagrams
</choice>
        </choices>
    </question>
```

```
 <question order="18">
        <id>18</id>
        <label>Dynamic Aspects Related To A System Are
Shown With Help Of</label>
        <difficulty>3</difficulty>
        <topics>
            <topic>UML Diagram</topic>
        </topics>
        <choices>
            <choice valid="flase"> Deployment Diagrams
</choice>
            <choice valid="true"> Interaction Diagrams </choice>
            <choice valid="false"> Sequence Diagrams </choice>
            <choice valid="flase"> Use Case Diagrams </choice>
        </choices>
    </question>
</questions>
```