## 5. Java : OOPS

What feature allows different methods to have the same name and arguments type, but a different implementation is called?

Pick **ONE** option

- ◯ overloading
- ◯ overriding
- ◯ Java does not permit methods with same name and type signature
- ◯ None of the above

## 6. Java - output

Given the following code, what's the output?

```
public class Element {

        private final int id;

        public Element(int id) {
                this.id = id;
        }

        public void updateId(int id) {
                this.id = id;
        }

        public int getId() {
                return id;
        }

        public void setId(int id) {
                this.id = id;
        }

        public static void main(String[] args) {
                Element elem = new Element(5);
                elem.updateId(10);
                System.out.println(elem.getId());

        }

}
```

Pick **ONE** option

- ◯ Compilation fails
- ◯ Exception is thrown at runtime
- ◯ 5
- ◯ 10

## 7. Java hierarchy - output

Given the following code, what's the output?

```
public class Comics {

	public void getOne() {
		Calvin calvin = new Calvin();
		calvin.getValue();
	}

}

public class Calvin {

	private Hobbes hobbes;

	public String getValue() {
		return hobbes.getValue();
	}

}

public class Hobbes {

	private String value;

	public String getValue() {
		value = "Calvin"; //Line 25
		return value;
	}

}

public class Main {

	public static void main(String[] args) {
		Comics comic = new Comics();
		comic.getOne();

	}

}
```

Pick **ONE** option

- ◯ The code run with no output
- ◯ "Calvin"
- ◯ An exception is thrown at runtime
- ◯ Compilations fails at line 25

## 8. Java varargs I

Given the following code, what's the output?

```
public class StringVarArgsTest {

        public void execute(int[]... values) {
                for (int[] val : values) {
                        System.out.print(val[0]);
                }
        }

        public static void main(String[] args) {
                int[] x = { 1, 2, 3 };
                int[] y = { 7, 8, 9 };
                StringVarArgsTest strTest = new StringVarArgsTest();
                strTest.execute(x, y);
        }

}
```

Pick **ONE** option

- ◯ Compilations fails
- ◯ Exception on runtime
- ◯ 1
- ◯ 17
- ◯ 123

## 9. Java varargs II

Given the following code, what's the output?

```
public class StringVarArgsTest {

        public void execute(String... values) {
                System.out.print(values[0] + " 1");

        }

        public void execute(String value) {
                System.out.print(value + " 2");
        }

        public static void main(String[] args) {
                StringVarArgsTest strTest = new StringVarArgsTest();
                strTest.execute("alpha");
        }

}
```

Pick **ONE** option

- ◯ alpha 1
- ◯ alpha 2
- ◯ alpha 2alpha 1
- ◯ alpha 1alpha 2

## 10. Java polymorphism

Chose the option (that replaces **[INSERT LINE HERE]**) that will raise a java.lang.ClassCastException?

```
public interface IGodFather {
}

public class Father implements IGodFather{
}

public class Daughter extends Father {
}

public class GrandDaughter extends Daughter {
}


public class Main {

        public static void main(String[] args) {
                Daughter d = new Daughter();
                // [INSERT LINE HERE]
        }

}
```

Pick **ONE** option

- ◯ Father f = d;
- ◯ IGodFather ifath = (Father) d;
- ◯ IGodFather igd = (GrandDaughter) d;
- ◯ Father dau = (Daughter) d;

## 11. Java concat

Given the following code, what's the output?

```
public class StringConcatTest {

        public static void main(String[] args) {
                String strTest = "987";
                strTest += "12";
                strTest.concat("34");
                strTest = strTest.concat("56");
                System.out.println(strTest);
        }

}
```

Pick **ONE** option

- 98712
- 98756
- 987123456
- 9871256
- 9871234

## 12. Java - output II

Given the following code, what's the output?

```
abstract class A {
        public A() {
                System.out.print("A");
        }
}

public class B extends A{
        public B() {
                System.out.print("B");
        }
}

public class C extends B {
        public C() {
                System.out.print("C");
        }
}

public class Test {

        public static void main(String[] args) {
                new C();
        }

}
```

Pick **ONE** option

- ◯ C
- ◯ ABC
- ◯ BC
- ◯ CBA
- ◯ Compilation fails

## 13. Java polymorphism II

Given the following code, what's the output?

```
public class Lamba {

	public String getValue() {
		return "Lamba";
	}
}

public class Alpha extends Lamba {

	public String getValue() {
		return "Alpha";
	}
}

public class Beta extends Alpha {

	public String getValue() {
		return super.getValue() + "Beta";
	}

	public static void main(String[] args) {
		new Beta().checkValues();
	}

	private void checkValues() {
		Alpha alpha = new Beta();
		Lamba lamba = new Alpha();
		Lamba lb = new Beta();
		System.out.print(alpha.getValue()+"-"+lamba.getValue()+"-"+lb.getValue());
	}

}
```

Pick **ONE** option

- ◯ AlphaBeta-Alpha-AlphaBeta
- ◯ Alpha-Lamba-Lamba
- ◯ Alpha-Lamba-AlphaBeta
- ◯ AlphaBeta-Alpha-Lamba
- ◯ Compilations fails