Object Oriented Programming

Quiz - 2

Prev Next

This quiz tests your knowledge of Java language semantics. Get a print out of this when you come to the class.

- First try answering without running the program.
- Then run the program to check if your understanding was indeed correct.
- Enter both the answers in the last column and submit.

Concept	Program	Your answer first, followed by the actual answer

Dynamic dispatch/ Method overloading/ Runtime Polymorphism	`// NTest.java`
	`public class NTest {`
	`  public static void main(String[] args) {`
	`    NTest nt = new NTest();`
	`    nt.method(2);`
	`    nt.method('2');`
	`  }`
	
	`  public void method(int i) { // Does nothing }`
	`  public void method(char c) { // Does nothing }`
	`}`	

Dynamic dispatch/ method overloading	`// OTest.java`
	`public class OTest {`
	`  public static void main(String[] args) {`
	`    OTest ot = new OTest();`
	`    ot.method(2);`
	`    ot.method(2);`
	`  }`
	
	`  public char method(int a) { return 'c'; }`
	`  public long method(int b) { return 0; }`
	`  /* Is it possible to overload methods that`
	`     differ only on the return type? */`
	`}`	

Instance method accessing static variable & calling static method	`// PTest.java`
	`public class PTest {`
	`  static int var;`
	`  public static void main(String[] args) {`
	`    PTest pt = new PTest();`
	`    pt.instanceMethod();`
	`  }`
	
	`  public void instanceMethod() {`
	`    var = 0;`
	`    staticMethod();`
	`    /* Is it appropriate to use this.var`
	`       or this.staticMethod()? */`
	`  }`
	
	`  public void staticMethod() {`
	`    System.out.println("static method";)`
	`  }`
	`}`	

Static method accessing instance variable & calling instance method	`// QTest.java`
	`public class QTest {`
	`  private int var;`
	`  public static void main(String[] args) {`
	`    var = 0;`
	`    instanceMethod();`
	`  }`
	
	`  public void instanceMethod() {`
	`    System.out.println("instance method");`
	`  }`
	`}`	

Working with null object	`// R.java`
	`public class R {`
	`    private int x;`
	`    public void set(int xi) { x = xi; }`
	`}`
	
	`// RTest.java`
	`public class RTest {`
	`  public static void main(String[] args) {`
	`    R r;`
	`    r.set(2);`
	`  }`
	`}`	

Working with null array	`// S.java`
	`public class S {`
	`    private int x;`
	`    public void set(int xi) { x = xi; }`
	`}`
	
	`// STest.java`
	`public class STest {`
	`  public static void main(String[] args) {`
	`    S[] s;`
	`    s.length;`
	`  }`
	`}`	

Working with null array element

```java
// T.java
public class T {
    private int x;
    public void set(int xi) { x = xi; }
}
```

```java
// TTest.java
public class TTest {
  public static void main(String[] args) {
    T[] t = new T[10];
    t[5].set(2);
  }
}
```

Crossing array boundary

```java
// U.java
public class U {
    private int x;
    public void set(int xi) { x = xi; }
    public int get() { return x; }
}
```

```java
// UTest.java
public class UTest {
  public static void main(String[] args) {
    U[] u = new U[10];
    for (int i=0; i<u.length; i++)
      u.set(i);
    System.out.println( u[10].get() );
  }
}
```

Default Constructor

```java
// V.java
public class V {
    private int x;
    public V(int xi) { x = xi; }
}
```

```java
// VTest.java
public class VTest {
  public static void main(String[] args) {
    V v1 = new V(5);
    V v2 = new V();
  }
}
```

Private constructor

```java
// W.java
public class W {
    private int x;
    private W() { x = 0; }
}
```

```java
// WTest.java
public class WTest {
  public static void main(String[] args) {
    W w = new W();
  }
}
```

Empty .java filename

```java
// .java
class X {   // Don't make it public
    private int x;
    public X(int xi) { x = xi; }
}
/* Compilation: javac .java
   Does it compile? Check it out. */
```

Static block with main

```java
// YTest.java
public class YTest {
  static{
    System.out.println("From static block");
  }

  public static void main(String args[]){
    System.out.println("Hello main");
  }
}
```

Static block without main

```java
// ZTest.java
public class ZTest {
  static{
    System.out.println("From static block");
    System.exit(0);
  }
}
```

In case you find any discrepancies please send them to swaminathanj@am.amrita.edu.