Seneca ​ College​

ofCHOOL​ ​Applied ​OF ​​COMPUTER ​ Arts​​ &​ ​​STUDIES ​​Technology

S ​

​ **Sample**​​ **JAC444**​​ **Midterm**​​ ​**Test** ​ ​**1** ​ **-**​​ **2016**​

**A.**​​**Theory**​(10​ ​marks​ ​=​ ​5​ ​marks​ +​​ ​5​ ​marks)

1. When​ ​can​ ​one​ ​implement​ ​a ​deep​​ ​cloning​ ​in​ ​Java?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.​ ​What​ ​are​ ​the​ ​differences​ ​between​ super​ ​and​ super()​ ​?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**B.**​​**Code**​​**–**​​**Quiz**​40​ ​marks​ ​=​ ​5​ ​quizzes​ ​\*​ ​8​ ​marks​ ​(3​ ​mark​ ​for​ ​correct ​ ​ answer​ ​and​ ​5​ for​​ ​explanation​**)**

1.​What ​​will​ ​happen ​​when​ ​you​ ​attempt​ ​to​ ​compile​ ​and​ ​run​ ​this​ ​code?

abstract ​ class​ ​ A​ ​ {​

​ ​abstract ​ public​ ​ void​ ​ method1();​

​ ​public​ ​void​ ​method2()​ ​{

​ System.out.println("The​ ​ second​ ​ method");​ ​ ​}

}

public​ class​ ​ B​ ​ extends​ ​ A​ ​ {​

​ ​public​ ​static​ ​void​ ​main(String​ ​argv[]){

​ A​ ​ e​ ​ =​ ​ new​ ​ B();​

​ e.​ ​ method2();​

​ ​}

​ ​public ​ void​ ​ method1(){​

​ ​System.out.println("The​ ​first​ ​method");

​ ​}

​ public​ ​ void​ ​ method2(){​

​ ​method1​ ​();

​ ​}

}

1)​ ​The​ ​code​ ​will​ ​compile​ ​and​ ​run,​ printing​ ​ ​out ​ the​ ​ words​ ​ "The​ ​ first​ ​ method"​

2)​ ​The​ ​compiler​ ​will​ ​complain ​ that​ ​ the​ ​ A​​ class​ ​ is​ ​ an​ ​ ​abstract ​ class.​

3)​ ​The ​ code​ ​ will​ ​ compile​ ​ ​and​ ​run,​ ​printing ​ out​​ the​ ​​words ​ "The​ ​ second​ ​ ​method ​ "​

4)​ ​The​ ​compiler​ ​will​ ​complain​ ​about​ ​the​ ​statement​ A​ ​e​ ​=​ ​new​ ​B();

# ANSWER:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Explanation***​:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**​

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

2. Given​ ​the​ ​following​ ​code,​ ​what​ ​will​ ​happen​ ​when ​you​​ ​try​ ​to​ ​compile​ ​and​ run​​ ​it?

public​ ​class​ ​Q2​ ​{ public​ ​static​ ​void ​ main(String[]​ ​ args)​ ​ {​

boolean​ ​b1​ ​=​ ​false;​ ​int​ ​val ​​= ​ 1​ ; if​ ​((b1​ ​==​ ​true) ​​&&​ ​((val​ ​+=​ ​1)​ ​== ​ 2))​ ​ System.out.println("Good:​ ​ ​"​ ​+​ ​val);

​ ​else

​ ​System.out.println("Bad:​ ​"​ ​+​ ​val); ​ ​}

}

A.​ ​Compilation​ ​error,​ ​attempting​ ​to​ ​perform​ ​binary​ ​comparison​ ​on​ ​logical​ ​data​ ​type.

B.​ ​Compilation​ ​and ​​output​ of​ ​ "Good:​ ​ 1​ ".

C.​ ​Compilation​ ​and​ ​output​ ​of ​ "Bad:​ ​ 2​ ".

D.​ ​Compilation​ ​and​ ​output​ ​of​ ​"Bad:​ ​1".

# ANSWER:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Explanation***​:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**​

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

3. Given​ ​the​ ​following​ ​code,​ ​what ​will​​ ​be​ ​the​ ​output?

class​ ​Int​ ​{

​ public​ ​ ​int​ ​i​ ​=​ ​1;

}

public​ ​class​ ​Q3​ ​{

​ public​ ​ static​ ​ void​ ​ main(String​ ​ argv[])​ {

Q3 ​ t​ ​ =​ ​ new​ ​ Q3();​

​ t.first();​

​ ​}

​ ​public​ ​void​ ​first() ​ {​

​ ​int​ ​i​ ​=​ ​2;

​ ​Int​ ​v​ ​=​ ​new​ ​Int();

4

​ v1.i​ ​ =​ ​ 3​ ;

​ ​second(v,​ ​i);

​ System.out.println(v.i);​

​ ​}

​ public​ ​ void​ ​ second(Int​ ​ v2,​​ int​ ​ i)​ ​ {​

​ ​i​ ​=​ ​0;

​ v2.i​ ​ =​ ​ 4​ ;

​ ​Int​ ​val​ =​ ​​new​ ​Int();

​ ​v2​ ​=​ ​val;

1

​ ​System.out.println(v.i​ ​+​ ​"​ ​"​ ​+​ ​i);

​ ​}

}

1. 1 ​ ​1

4

1. 1 0

1

1. 4 0

4

1. 1 0

4

# ANSWER:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Explanation***​:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**​

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

4. Given​ ​the​ ​following​ ​code,​ ​what ​will​​ ​be​ ​the​ ​output?

public​ ​class​ ​Q4​ ​{ public​ ​static​ ​void​ ​main(String​ ​args[])​ ​{

System.out.println('e' ​​-​ 'b'​​ ​+​ ​"A"​ ​+​ ​4); }

}

A.​ ​3A4

B. ​ 'e'​ ​ -​ ​ 'b'​ ​ +​ ​ "A"​ ​ ​+ ​ 4​

C.​ ​Compilation​ ​error

D.​ ​None​ ​of​ ​these

# ANSWER:\_A\_\_\_\_\_\_\_\_\_\_\_

***Explanation***​:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**​

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

What​ ​will​ ​happen​ ​when​ ​you​ ​try​ ​to​ ​compile​ ​and​ run​ ​ the​ ​ following​ ​ code?​ public​ ​class​ ​Q5​ ​{

public​ ​static ​ void​ ​ test()​ ​ {​

for​ ​(int​ ​i​ ​=​ ​0;​ ​i​ ​<​ ​3;​ ​i++)

System.out.print(i);

System.out.print(i);

}

public​ ​static​ ​void​ ​main(String[]​ ​args)​ ​{ test();

}

}

1. 0122
2. 0123
3. Compilation​ ​error
4. None ​ of​ ​ these​

# ANSWER:\_\_\_\_C\_\_\_\_\_\_\_\_\_\_\_

***Explanation***​:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**​

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**C.**​​**What**​​**does**​​**the**​​**following**​​**code**​​**print?**​​**Please**​​**explain** ​ **your**​​ ​**answer.**  ​20 ​ marks​ ​ =​ ​ 5​ ​ marks​ ​ +​ ​ 15​ ​​marks

class​ ​X​ ​{ public​ ​void​ ​m1()​ ​{

System.out.print("X​ ​m1()​ ​-​ ​"); }

public​ ​void​ ​m2()​ ​{

System.out.print("X​ ​m2()​ ​-​ ​"); }

}

public​ ​class​ ​Y​ ​extends​ ​X ​​{

public​ ​void​ ​m1()​ ​{

System.out.print("Y​ ​m1()​ ​-​ ​");

super.m2();

}

public​ ​static​ ​void​ ​main(String​ ​args[])​ ​{ X​ ​x​ ​=​ ​new​ ​Y();

x.m1();

}

​ ​}

***Output:***

**\_\_\_\_\_\_\_\_\_Y m1() –X m1() -\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Explanation***​:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**​

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

class​ ​Dog​ ​{ public​ ​void​ ​bark()​ ​{

System.out.println("Dog​ ​-​ ​bark​ ​-​ ​"); }

}

class​ ​MyDog​ ​extends ​ Dog​ ​ {​ public​ ​void​ ​bark() ​ {​

String​ ​s1​ ​=​ ​new​ ​String("woofer"); String​ ​s2 ​ =​ ​ new​ ​ String("woofer");​ if ​ (​ s1 ​​==​ ​s2)

​ ​System.out.println("Same ​ ");​ if​ ​(s1.equals(s2))

​ ​System.out.println("Equals​ ​");

System.out.println("woofer​ ​-​ ​bark​ ​-​ ​"); }

}

public​ ​class​ ​Bark​ ​{

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

Dog​ ​woofer ​ =​​ ​new​ Dog();​ Dog ​​nipper​ ​=​ ​new​ ​MyDog(); woofer.bark(); nipper.bark();

}

​ ​}

***Output:***

**\_\_\_\_\_** Dog​ ​-​ ​bark​ ​-​**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_** woofer​ ​-​ ​bark​ ​**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Explanation***​:

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**D.**​​**Code**​​**–**​​**Development**​(Question​ ​1​ ​=​ ​10​ ​marks​ ​Question​ ​2​ ​=​ ​20marks)

1. ​ Consider​ ​ the​ ​ following​ ​ Java​ ​ expression​

x​ ​->​ ​{​ ​return​ ​(x​ ​>​ ​0​ ​&&​ ​x​ ​<​ 10)​ ;​ ​}

If ​ this​ ​ is​ ​ a​ ​ valid​ ​ expression​ ​ in​ ​ Java,​ ​explain​ ​ ​what ​it​​ ​does​ ​and ​​how​ ​would​ ​you​ ​use​ ​it.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Write ​ a​ ​ Java​ ​ program​ ​ that​​ takes​ ​ two​ ​ arguments​ ​ on​ ​ a​ ​ command​ ​ line.​ ​ The​​ ​first ​ ​argument ​ is​ ​ ​the​ ​name ​ ​of the​ ​text​ ​file,​ ​and​ ​the​ ​second ​ is​ ​ a​ ​string.​ ​ Your​ ​ program​ ​ must​ ​ read​​ the​​ ​file ​and​ ​​print ​ all​ ​ ​the​ ​lines​ from​ ​ the​ file​ where​​ ​the ​​given​ ​string​ ​is​ ​found.

Write​ ​another​ ​Java​ ​program​ ​that​ ​takes​ ​many​ ​arguments ​ on​ ​ a​ ​ command​​ line.​ ​ All​ ​ arguments​ ​ are​​ ​the ​ text​ file​ ​names, ​ except​ ​ the​ ​ last,​ ​ which​ ​ ​is​ ​a ​​string.​ Find​ ​ how​ ​​many​ ​lines​ ​in​ ​a​ ​file​ contain​​ ​the​ ​given ​​string. Every​ ​file​ ​must​ ​be​ ​read​ ​in​ ​a​ ​different​ ​thread.

*Use*​​*in*​​*the*​​*second*​​*program*​​*as*​​*much*​​*code*​​*as*​​*you*​​*can*​​*from*​​*first*​​*program.*​​*Please*​​*properly*​​*document*​​*your*​​*code.*