



United International University (UIU)
Dept. of Computer Science and Engineering (CSE)

MID TERM EXAM :: Spring 2019

Course Code: **CSI 211** Course Title: **Object-Oriented Programming**

Date: **7/3/19** Total Marks: **30** Time: **1 Hour 45 mins**

1.	<p>Write the output of the following code:</p> <table border="1" style="width: 100%; border-collapse: collapse;"><tr><td style="width: 50%; padding: 5px;"><pre>public class Zoo { public static void main(String[] args) { Animal a = new Animal(); Bird b = new Magpie(); Magpie c = (Magpie) b; a.fly(); b.fly(); ((Magpie)b).fly(15); c.fly(); a.eat(); b.eat(); c.eat(); } }</pre></td><td style="width: 50%; padding: 5px;"><pre>class Magpie extends Bird{ Magpie(){ System.out.println("A magpie is created."); } void fly(){ System.out.println("Magpie is flying..."); } void fly(int speed){ System.out.println("Magpie is flying at speed: " + speed); } void eat(){ System.out.println("Magpie is eating."); } }</pre></td></tr><tr><td style="width: 50%; padding: 5px;"><pre>class Bird extends Animal{ Bird(){ System.out.println("A bird is created."); } void fly(){ System.out.println("Flying..."); } }</pre></td><td style="width: 50%; padding: 5px;"><pre>class Animal{ Animal() { System.out.println("An animal is created."); } void fly(){ System.out.println("Don't know if I can fly!"); } void eat(){ System.out.println("Eating..."); } }</pre></td></tr></table>	<pre>public class Zoo { public static void main(String[] args) { Animal a = new Animal(); Bird b = new Magpie(); Magpie c = (Magpie) b; a.fly(); b.fly(); ((Magpie)b).fly(15); c.fly(); a.eat(); b.eat(); c.eat(); } }</pre>	<pre>class Magpie extends Bird{ Magpie(){ System.out.println("A magpie is created."); } void fly(){ System.out.println("Magpie is flying..."); } void fly(int speed){ System.out.println("Magpie is flying at speed: " + speed); } void eat(){ System.out.println("Magpie is eating."); } }</pre>	<pre>class Bird extends Animal{ Bird(){ System.out.println("A bird is created."); } void fly(){ System.out.println("Flying..."); } }</pre>	<pre>class Animal{ Animal() { System.out.println("An animal is created."); } void fly(){ System.out.println("Don't know if I can fly!"); } void eat(){ System.out.println("Eating..."); } }</pre>	[5]
<pre>public class Zoo { public static void main(String[] args) { Animal a = new Animal(); Bird b = new Magpie(); Magpie c = (Magpie) b; a.fly(); b.fly(); ((Magpie)b).fly(15); c.fly(); a.eat(); b.eat(); c.eat(); } }</pre>	<pre>class Magpie extends Bird{ Magpie(){ System.out.println("A magpie is created."); } void fly(){ System.out.println("Magpie is flying..."); } void fly(int speed){ System.out.println("Magpie is flying at speed: " + speed); } void eat(){ System.out.println("Magpie is eating."); } }</pre>					
<pre>class Bird extends Animal{ Bird(){ System.out.println("A bird is created."); } void fly(){ System.out.println("Flying..."); } }</pre>	<pre>class Animal{ Animal() { System.out.println("An animal is created."); } void fly(){ System.out.println("Don't know if I can fly!"); } void eat(){ System.out.println("Eating..."); } }</pre>					
2.	<p>Consider the following FruitShop class. Now, write a class Fruit in such a way that FruitShop class will give expected output as shown below.</p> <table border="1" style="width: 100%; border-collapse: collapse;"><thead><tr><th style="width: 50%; text-align: left; padding: 5px;">Class FruitShop</th><th style="width: 50%; text-align: left; padding: 5px;">Output</th></tr></thead><tbody><tr><td style="width: 50%; padding: 5px;"><pre>public class FruitShop { public static void main(String[] args) { Fruit fruit1 = new Fruit("Apple", 3.5, 110); Fruit fruit2 = new Fruit("Mango", 5, 90); fruit1.reducePricePerKG(10); fruit2.increasePricePerKG(20); fruit1.printDetails(); fruit2.printDetails(); } }</pre></td><td style="width: 50%; padding: 5px;"><pre>Fruit Details: Name: Apple Weight: 3.5kg Price per kg: 100.0 Total price: 350.0 Fruit Details: Name: Mango Weight: 5.0kg Price per kg: 110.0 Total price: 550.0</pre></td></tr></tbody></table>	Class FruitShop	Output	<pre>public class FruitShop { public static void main(String[] args) { Fruit fruit1 = new Fruit("Apple", 3.5, 110); Fruit fruit2 = new Fruit("Mango", 5, 90); fruit1.reducePricePerKG(10); fruit2.increasePricePerKG(20); fruit1.printDetails(); fruit2.printDetails(); } }</pre>	<pre>Fruit Details: Name: Apple Weight: 3.5kg Price per kg: 100.0 Total price: 350.0 Fruit Details: Name: Mango Weight: 5.0kg Price per kg: 110.0 Total price: 550.0</pre>	[5]
Class FruitShop	Output					
<pre>public class FruitShop { public static void main(String[] args) { Fruit fruit1 = new Fruit("Apple", 3.5, 110); Fruit fruit2 = new Fruit("Mango", 5, 90); fruit1.reducePricePerKG(10); fruit2.increasePricePerKG(20); fruit1.printDetails(); fruit2.printDetails(); } }</pre>	<pre>Fruit Details: Name: Apple Weight: 3.5kg Price per kg: 100.0 Total price: 350.0 Fruit Details: Name: Mango Weight: 5.0kg Price per kg: 110.0 Total price: 550.0</pre>					

3	<p>a. Consider the following classes. Will the code run without errors? If so, please specify the errors. Otherwise, write the output of the code.</p> <table border="1" data-bbox="240 294 1299 634"> <tr> <td data-bbox="246 302 786 630"> <pre>class B{ void a(){ System.out.println("I am first A inside B"); } } class C extends B{ void a(int a){ System.out.println("I am second a inside C, my val is " + a); } }</pre> </td> <td data-bbox="792 302 1292 630"> <pre>public class D{ public static void main(String[] args) { B obj = new C(); obj.a(); obj.a(5); } }</pre> </td> </tr> </table>	<pre>class B{ void a(){ System.out.println("I am first A inside B"); } } class C extends B{ void a(int a){ System.out.println("I am second a inside C, my val is " + a); } }</pre>	<pre>public class D{ public static void main(String[] args) { B obj = new C(); obj.a(); obj.a(5); } }</pre>	[4]
<pre>class B{ void a(){ System.out.println("I am first A inside B"); } } class C extends B{ void a(int a){ System.out.println("I am second a inside C, my val is " + a); } }</pre>	<pre>public class D{ public static void main(String[] args) { B obj = new C(); obj.a(); obj.a(5); } }</pre>			
	b. Is it a must for abstract classes to have abstract method/methods?	[1]		
4	<p>a. Write the output of the following code:</p> <table border="1" data-bbox="240 806 1338 1285"> <tr> <td data-bbox="246 814 808 1276"> <pre>public class Course { String name, id; int credit; public Course(String name, String id, int credit) { this.name = name; this.id = id; this.credit = credit; } public void display() { System.out.printf("%s-%s-%d\n", name, id, credit); } public void updateCourse(Course c1){ this.name = c1.name; this.id = c1.id; c1.credit--; } }</pre> </td> <td data-bbox="815 814 1331 1276"> <pre>public class Application { public static void main(String[] args) { Course c1=new Course("OOP","CSI211",3); Course c2=new Course("SPL","CSI121",1); c1.display(); c2.display(); c1.updateCourse(c2); c1.display(); c2.display(); } }</pre> </td> </tr> </table>	<pre>public class Course { String name, id; int credit; public Course(String name, String id, int credit) { this.name = name; this.id = id; this.credit = credit; } public void display() { System.out.printf("%s-%s-%d\n", name, id, credit); } public void updateCourse(Course c1){ this.name = c1.name; this.id = c1.id; c1.credit--; } }</pre>	<pre>public class Application { public static void main(String[] args) { Course c1=new Course("OOP","CSI211",3); Course c2=new Course("SPL","CSI121",1); c1.display(); c2.display(); c1.updateCourse(c2); c1.display(); c2.display(); } }</pre>	[3]
<pre>public class Course { String name, id; int credit; public Course(String name, String id, int credit) { this.name = name; this.id = id; this.credit = credit; } public void display() { System.out.printf("%s-%s-%d\n", name, id, credit); } public void updateCourse(Course c1){ this.name = c1.name; this.id = c1.id; c1.credit--; } }</pre>	<pre>public class Application { public static void main(String[] args) { Course c1=new Course("OOP","CSI211",3); Course c2=new Course("SPL","CSI121",1); c1.display(); c2.display(); c1.updateCourse(c2); c1.display(); c2.display(); } }</pre>			
	<p>b. Fix the following code. You cannot remove any lines. You can only add or edit existing lines.</p> <table border="1" data-bbox="240 1394 1338 1864"> <tr> <td data-bbox="246 1402 808 1856"> <pre>package zoo; class Animal { String name; protected int legs; Animal(String name, int legs){ this.name = name; this.legs = legs; } void print(){ System.out.println(name + " " + legs); } }</pre> </td> <td data-bbox="815 1402 1331 1856"> <pre>package zoo.birds; class Bird extends Animal{ String color; Bird(){ } Bird(String name, int legs, String color){ this(name, legs); super.color = color; } void print(){ System.out.println(name + " " + legs + " " + color); } }</pre> </td> </tr> </table>	<pre>package zoo; class Animal { String name; protected int legs; Animal(String name, int legs){ this.name = name; this.legs = legs; } void print(){ System.out.println(name + " " + legs); } }</pre>	<pre>package zoo.birds; class Bird extends Animal{ String color; Bird(){ } Bird(String name, int legs, String color){ this(name, legs); super.color = color; } void print(){ System.out.println(name + " " + legs + " " + color); } }</pre>	[2]
<pre>package zoo; class Animal { String name; protected int legs; Animal(String name, int legs){ this.name = name; this.legs = legs; } void print(){ System.out.println(name + " " + legs); } }</pre>	<pre>package zoo.birds; class Bird extends Animal{ String color; Bird(){ } Bird(String name, int legs, String color){ this(name, legs); super.color = color; } void print(){ System.out.println(name + " " + legs + " " + color); } }</pre>			

5.	<p>a. Will the code run without errors? If so, please specify the errors. Otherwise, write the output of the following code.</p> <table border="1" data-bbox="240 294 1344 655"> <tr> <td data-bbox="240 294 779 655"> <pre>class CrazyVars{ static int i; int j; void incrementI(){ CrazyVars.i++; } void incrementJ(){ this.j++; } }</pre> </td> <td data-bbox="779 294 1344 655"> <pre>public class Execution{ public static void main(String[] args){ CrazyVars workerOne = new CrazyVars(); CrazyVars workerTwo = new CrazyVars(); workerOne.incrementI(); workerTwo.incrementJ(); workerTwo.incrementI(); workerOne.incrementJ(); System.out.println(workerOne.i + workerTwo.j); } }</pre> </td> </tr> </table>	<pre>class CrazyVars{ static int i; int j; void incrementI(){ CrazyVars.i++; } void incrementJ(){ this.j++; } }</pre>	<pre>public class Execution{ public static void main(String[] args){ CrazyVars workerOne = new CrazyVars(); CrazyVars workerTwo = new CrazyVars(); workerOne.incrementI(); workerTwo.incrementJ(); workerTwo.incrementI(); workerOne.incrementJ(); System.out.println(workerOne.i + workerTwo.j); } }</pre>	[2.5]
<pre>class CrazyVars{ static int i; int j; void incrementI(){ CrazyVars.i++; } void incrementJ(){ this.j++; } }</pre>	<pre>public class Execution{ public static void main(String[] args){ CrazyVars workerOne = new CrazyVars(); CrazyVars workerTwo = new CrazyVars(); workerOne.incrementI(); workerTwo.incrementJ(); workerTwo.incrementI(); workerOne.incrementJ(); System.out.println(workerOne.i + workerTwo.j); } }</pre>			
	<p>b. Explain what is wrong with this code. Fix the code. You CANNOT add/delete any line.</p> <table border="1" data-bbox="240 735 1344 940"> <tr> <td data-bbox="240 735 876 940"> <pre>interface ABC { String s = "123"; void method1(int a); void greeter(); }</pre> </td> <td data-bbox="876 735 1344 940"> <pre>class C implements ABC{ String s = "new"; public void greeter() { System.out.println("Hi"); } }</pre> </td> </tr> </table>	<pre>interface ABC { String s = "123"; void method1(int a); void greeter(); }</pre>	<pre>class C implements ABC{ String s = "new"; public void greeter() { System.out.println("Hi"); } }</pre>	[2.5]
<pre>interface ABC { String s = "123"; void method1(int a); void greeter(); }</pre>	<pre>class C implements ABC{ String s = "new"; public void greeter() { System.out.println("Hi"); } }</pre>			
6.	<p>a. Write the output of the following code:</p> <table border="1" data-bbox="240 1029 1344 1415"> <tr> <td data-bbox="240 1029 779 1415"> <pre>class SwapFunc { static void swapReference(RefFunc a, RefFunc b) { RefFunc c = a; a = b; b = c; } }</pre> </td> <td data-bbox="779 1029 1344 1415"> <pre>public class RefFunc { int a; int b; public RefFunc(int a, int b) { this.a = a; this.b = b; } public static void main(String[] args) { RefFunc obj1 = new RefFunc(5, 6); RefFunc obj2 = new RefFunc(7, 8); SwapFunc.swapReference(obj1, obj2); System.out.println(obj1.a + "\t" + obj2.b); } }</pre> </td> </tr> </table>	<pre>class SwapFunc { static void swapReference(RefFunc a, RefFunc b) { RefFunc c = a; a = b; b = c; } }</pre>	<pre>public class RefFunc { int a; int b; public RefFunc(int a, int b) { this.a = a; this.b = b; } public static void main(String[] args) { RefFunc obj1 = new RefFunc(5, 6); RefFunc obj2 = new RefFunc(7, 8); SwapFunc.swapReference(obj1, obj2); System.out.println(obj1.a + "\t" + obj2.b); } }</pre>	[2.5]
<pre>class SwapFunc { static void swapReference(RefFunc a, RefFunc b) { RefFunc c = a; a = b; b = c; } }</pre>	<pre>public class RefFunc { int a; int b; public RefFunc(int a, int b) { this.a = a; this.b = b; } public static void main(String[] args) { RefFunc obj1 = new RefFunc(5, 6); RefFunc obj2 = new RefFunc(7, 8); SwapFunc.swapReference(obj1, obj2); System.out.println(obj1.a + "\t" + obj2.b); } }</pre>			
	<p>b. Will the code run without errors? If so, please specify the errors. Otherwise, write the output of the following code.</p> <table border="1" data-bbox="240 1533 1344 1919"> <tr> <td data-bbox="240 1533 779 1919"> <pre>abstract class Bike { int brandType; private abstract void velocity(); abstract void showType() { System.out.println("brandType is " + brandType); } public void firstCall() { System.out.println("A bike is being made"); } }</pre> </td> <td data-bbox="779 1533 1344 1919"> <pre>public class Runner extends Bike { protected void firstCall() { super.firstCall(); System.out.println("A Runner is being made"); } private void velocity() { System.out.println("Max velocity is 125"); } public static void main(String[] args) { Bike obj = new Runner(); obj.firstCall(); } }</pre> </td> </tr> </table>	<pre>abstract class Bike { int brandType; private abstract void velocity(); abstract void showType() { System.out.println("brandType is " + brandType); } public void firstCall() { System.out.println("A bike is being made"); } }</pre>	<pre>public class Runner extends Bike { protected void firstCall() { super.firstCall(); System.out.println("A Runner is being made"); } private void velocity() { System.out.println("Max velocity is 125"); } public static void main(String[] args) { Bike obj = new Runner(); obj.firstCall(); } }</pre>	[2.5]
<pre>abstract class Bike { int brandType; private abstract void velocity(); abstract void showType() { System.out.println("brandType is " + brandType); } public void firstCall() { System.out.println("A bike is being made"); } }</pre>	<pre>public class Runner extends Bike { protected void firstCall() { super.firstCall(); System.out.println("A Runner is being made"); } private void velocity() { System.out.println("Max velocity is 125"); } public static void main(String[] args) { Bike obj = new Runner(); obj.firstCall(); } }</pre>			

--	--	--