WEEK 5 – INHERITANCE MCQS

Question 1

Complete

Marked out of 1.00

☐ ⚐ Flag question

Which of the following statements are incorrect?

○ a. private members of class can only be accessed by other members of the class.

○ b. public members of class can be accessed by any code in the program.

◉ c. protected members of a class can be inherited by a sub class, and become private members of the sub class.

○ d. private members of class can be inherited by a sub class, and become protected members in sub class.

Question 2

Complete

Marked out of 1.00

☐ ⚐ Flag question

Given the following code, which is the simplest print statement that can be inserted into the print() method?

```
// Filename: MyClass.java
public class MyClass extends MySuperclass {
    public static void main(String[] args) {
        MyClass object = new MyClass();
        object.print();
    }
    public void print() {
        // INSERT CODE HERE THAT WILL PRINT
        // THE "Hello, world!" STRING FROM THE Message
        // CLASS.
    }
}
class MySuperclass {
    Message msg = new Message();
}
class Message {
    // The message that should be printed:
    String text = "Hello, world!";
}
```

○ a. System.out.println(object.msg.text);

◉ b. System.out.println(msg.text);

○ c. System.out.println(Message.text);

○ d. System.out.println(super.msg.text);

Question **3**

Complete

Marked out of
1.00

☐⚑ Flag
question

Given the following code, which of these constructors can be added to MySub class without causing a compile-time error?

```
class MySuper {
    int number;
    MySuper(int i) { number = i; }
}
class MySub extends MySuper {
    int count;
    MySub(int cnt, int num) {
        super(num);
        count=cnt;
    }
    // INSERT ADDITIONAL CONSTRUCTOR HERE
}
```

- ○ a. MySub(int cnt) { super(cnt); this(cnt, 0); }
- ○ b. MySub() {}
- ◉ c. MySub(int cnt) { this(cnt, cnt); }
- ○ d. MySub(int cnt) { count = cnt; super(cnt); }

Question **4**

Complete

Marked out of
1.00

☐ ⚑ Flag
question

1. public class TestPoly {

2.   public static void main(String [] args){

3.   Parent p = new Child();

4.   }

5. }

6.

7. class Parent {

8.   public Parent() {

9.   super();

10.   System.out.println("instantiate a parent");

11. }

12. }

13.

14. class Child extends Parent {

15. public Child() {

16.   System.out.println("instantiate a child");

17. }

18. }

What is the result?

- ○ a. instantiate a child
- ◉ b. instantiate a parent

     instantiate a child
- ○ c. instantiate a parent
- ○ d. instantiate a child

     instantiate a parent

Question **5**

Complete

Marked out of
1.00

☐ ⚑ Flag
question

Which of the following modifiers cannot be applied to a top level class?

- ○ a. final
- ◉ b. private
- ○ c. public
- ○ d. abstract

Consider the following class heirarchies

class A { }

class B extends A { }

class C extends B { }

And the following method declaration

public B doSomething 🍔 {

   // some valid code fragments

   return xx;

}

Objects of which class (from the heirarchy shown above) can be safely substituted in place of xx in the method doSomething 🍔 ?

  ○ a. An array object of class B

  ○ b. Object of class A

  ◉ c. Object of class C

  ○ d. An array object of class C

Question **7**

Complete

Marked out of
1.00

☐ ▷ Flag
question

Given the following,

1. class MySuper {

2.   public MySuper(int i) {

3.   System.out.println("super " + i);

4.   }

5. }

6.

7. public class MySub extends MySuper {

8.   public MySub() {

9.   super(2);

10.   System.out.println("sub");

11. }

12.

13. public static void main(String [] args) {

14.   MySuper sup = new MySub();

15. }

16. }

What is the result?

○ a. Compilation fails at line 9.

○ b. sub

        super 2

⦿ c. super 2

        sub

○ d. Compilation fails at line 14.

Question **8**

Complete

Marked out of
1.00

☐ ▷ Flag
question

What type of inheritance does Java have?

○ a.   double inheritance

○ b.   class inheritance

○ c.   multiple inheritance

⦿ d.   single inheritance

Question **9**

Complete

Marked out of
1.00

Flag
question

What restriction is there on using the super reference in a constructor?

- a. It must be used in the last statement of the constructor.

- b. It can only be used in the parent's constructor.

- c. Only one child class can use it.

- d. It must be used in the first statement of the constructor.

Question **10**

Complete

Marked out of
1.00

Flag
question

Can an object of a child type be assigned to a variable of the parent type? For example,

Card crd;

BirthDay bd = new BirthDay("Lucinda", 42);

crd = bd; // is this correct?

- a. No-there must always be an exact match between the variable and the object types.

- b. Yes-any object can be assigned to any reference variable.

- c. Yes-an object can be assigned to a reference variable of the parent type.

- d. No-but a object of parent type can be assigned to a variable of child type.

Question **11**

Complete

Marked out of
1.00

☐🏳 Flag
question

Given the following:

1. public class MyClass {

2.   public static void main(String[] args) {

3.   Derived d = new Derived("hello");

4.   }

5. }

6.

7. class Base {

8.   Base() { this("a", "b"); }

9.

10.   Base(String x, String y) { System.out.println(x + y); }

11. }

12.

13. class Derived extends Base {

14.   Derived(String s) { System.out.println(s); }

15. }

What is the output?

- ○ a.   It will print ab
- ○ b.   It will print hello followed by ab.
- ○ c.   It will print hello.
- ⊙ d.   It will print ab followed by hello.

Question **12**

Complete

Marked out of
1.00

☐🏳 Flag
question

What restriction is there on using the super reference in a constructor?

- ○ a.   It must be used in the last statement of the constructor.
- ○ b.   Only one child class can use it.
- ○ c.   It can only be used in the parent's constructor.
- ⊙ d.   It must be used in the first statement of the constructor.

Question **13**

Complete

Marked out of
1.00

☐🏳 Flag
question

Which statement is true?

- ⊙ a.   If both a subclass and its superclass do not have any declared constructors, the implicit default constructor of the subclass will call super() when run.
- ○ b.   If super() is the first statement in the body of a constructor, then this() can be declared as the second statement.
- ○ c.   A super() or this() call must always be provided explicitly as the first statement in the body of a constructor.
- ○ d.   If neither super() nor this() is declared as the first statement in the body of a constructor, then this() will implicitly be inserted as the first statement.

Question **14**

Complete

Marked out of
1.00

☐🏳 Flag
question

A class Animal has a subclass Mammal. Which of the following is true:

- ○ a.   Because of single inheritance, Mammal can have no subclasses.
- ○ b.   Because of single inheritance, Animal can have only one subclass.
- ○ c.   Because of single inheritance, Mammal can have no siblings.
- ⊙ d.   Because of single inheritance, Mammal can have no other parent than Animal.

Given the following code:

```
class B { int m = 7; }
class D extends B { int m = 9; }
public class TestBaseDerived {
    public static void main(String[] args) {
        B b = new B();
        D d = new D();
        B bd = new D();
        System.out.printf("%d %d %d", b.m, d.m, bd.m);
    }
}
```

What will be the output on executing the above code ?

- ○ a. 9 7 9
- ◉ b. 7 9 7
- ○ c. 9 9 7
- ○ d. 7 9 9

Which statement is true about the use of modifiers?

- ◉ a. You cannot specify accessibility of local variables. They are only accessible within the block in which they are declared.
- ○ b. Subclasses of a class must reside in the same package as the class they extend.
- ○ c. Local variables can be declared static.
- ○ d. If no accessibility modifier (public, protected, and private) is specified for a member declaration, the member is only accessible for classes in the same package and subclasses of its class in any package.

Which one of the following statement is false?

- ○ a. A final class cannnot be abstract.
- ○ b. A top level class in which all the members are declared private, can be declared public.
- ◉ c. All members of the superclass are inherited by the subclass.
- ○ d. The subclass of a non-abstract class can be declared abstract.

Question **18**

Complete

Marked out of
1.00

☐▷ Flag
question

Given the following code, which is the simplest print statement that can be inserted into the print() method?

```
// Filename: MyClass.java
public class MyClass extends MySuperclass {
 public static void main(String[] args) {
  MyClass object = new MyClass();
  object.print();
 }

 public void print() {
  // INSERT CODE HERE THAT WILL PRINT
  // THE "Hello, world!" STRING FROM THE Message
  // CLASS.
 }
}

class MySuperclass {
 Message msg = new Message();
}

class Message {
 // The message that should be printed:
 String text = "Hello, world!";
}
```

- ○ a. System.out.println(msg.text);
- ○ b. System.out.println(Message.text);
- ○ c. System.out.println(object.msg.text);
- ◉ d. System.out.println(super.msg.text);

Question 19

Complete

Marked out of
1.00

☐🚩 Flag
question

Analyse the following 2 classes and select the correct statement.

```
class A {

 private int x = 0;

 static int y = 1;

 protected int q = 2;

}


class B extends A {

 void method() {

  System.out.println(x);

  System.out.println👆 ;

  System.out.println(q);

 }

}
```

- a. The code fails to compile because you can't subclass a class with protected variables.
- b. The code compiles correctly, and the following is displayed:012
- ● c. The code fails to compile because the variable x is not available to class B.
- d. The code fails to compile because you can't subclass a class with static variables.

Question **20**

Complete

Marked out of
1.00

☐⚑ Flag
question

Given the following:

class Vehicle { }

class FourWheeler extends Vehicle { }

class Car extends FourWheeler { }

public class TestVehicle

{

   public static void main(String[] args)

   {

      Vehicle v = new Vehicle();

      FourWheeler f = new FourWheeler();

      Car c = new Car();

      xxxxxxx

   }

}

Which of the following statement is legal, which can be substituted for xxxxxxx?

○ a. c = v;

○ b. c = f;

○ c. f = v;

◉ d. v = c;

Given the following,

1. class B extends A {
2.   int getID() {
3.   return id;
4.   }
5. }
6. class C {
7.   public int name;
8. }
9. class A {
10.   C c = new C();
11.   public int id;
12. }

Which one is correct about instances of the classes listed above?

- ○ a.   A is-a B
- ⦿ b.   B has-a C
- ○ c.   C is-a A
- ○ d.   B has-a A

Say that there are three classes: Computer, AppleComputer, and IBMComputer. What are the likely relationships between these classes?

- ⦿ a.   Computer is the superclass, AppleComputer and IBMComputer are subclasses of Computer.
- ○ b.   Computer is a superclass, AppleComputer is a subclasses of Computer, and IBMComputer is a subiclas of AppleComputer
- ○ c.   Computer, AppleComputer and IBMComputer are sibling classes.
- ○ d.   IBMComputer is the superclass, AppleComputer and Computer are subclasses of IBMComputer.

Given a method in a class, what access modifier do you use to restrict access to that method to only the other members of the same class?

- ○ a.   static
- ○ b.   volatile
- ○ c.   protected
- ⦿ d.   private

Which statement is true?

○ a. A final class can be extended by any number of classes

○ b. Inheritance defines a has-a relationship between a superclass and its subclasses.

○ c. Every Java object has a public method named length.

◉ d. Every Java object has a public method named equals.

Assuming Card is the base class of Valentine, Holiday and Birthday, in order for the following code to be correct, what must be the type of the reference variable card

```
_____ card;
card = new Valentine( "Joe", 14 ) ;
card.greeting();
card = new Holiday( "Bob" ) ;
card.greeting();
card = new Birthday( "Emily", 12 ) ;
card.greeting();
```

○ a. Birthday

◉ b. Card

○ c. Valentine

○ d. Holiday

The concept of inheritance provides the idea of

○ a. reusability

◉ b. all of these

○ c. Taking more than one form

○ d. data hiding

Which one of the following statement is false?

◉ a. All members of the superclass are inherited by the subclass.

○ b. A final class cannnot be abstract.

○ c. A top level class in which all the members are declared private, can be declared public.

○ d. The subclass of a non-abstract class can be declared abstract.

Marked out of
1.00

☐⚑ Flag
question

Which of the following modifiers can be applied to a constructor?

- ⦿ a. protected
- ○ b. synchronized
- ○ c. static
- ○ d. transient

Question 29

Complete

Marked out of
1.00

☐⚑ Flag
question

Which of the following is correct syntax for defining a new class Jolt based on the superclass SoftDrink?

- ○ a. class Jolt defines SoftDrink { //additional definitions go here }
- ○ b. class Jolt implements SoftDrink { //additional definitions go here }
- ○ c. class Jolt isa SoftDrink { //additional definitions go here }
- ⦿ d. class Jolt extends SoftDrink { //additional definitions go here }

Question 30

Complete

Marked out of
1.00

☐⚑ Flag
question

class A { A(int i) {} }   // 1

class B extends A { }   // 2

Which one of the following statements is correct?

- ○ a. Compile-time error at 1.
- ⦿ b. Compile-time error at 2.
- ○ c. compiler attempts to create a default constructor for class A.
- ○ d. Compiles successfully without any errors.

Question 31

Complete

Marked out of
1.00

☐⚑ Flag
question

Say that class Rodent has a child class Rat and another child class Mouse. Class Mouse has a child class PocketMouse. Examine the following

Rodent rod;

Rat rat = new Rat();

Mouse mos = new Mouse();

PocketMouse pkt = new PocketMouse();

Which of the following array declarations is correct for an array that is expected to hold up to 10 objects of types Rat, Mouse, and PocketMouse?

- ○ a. Rat[] array = new Rat[10];
- ○ b. Rodent[] array = new Rat[10];
- ○ c. Rodent[10] array;
- ⦿ d. Rodent[] array = new Rodent[10];

Say that class Rodent has a child class Rat and another child class Mouse. Class Mouse has a child class PocketMouse. Examine the following

Rodent rod;

Rat rat = new Rat();

Mouse mos = new Mouse();

PocketMouse pkt = new PocketMouse();

Which one of the following will cause a compiler error?

◉ a. pkt = rat;

○ b. pkt = null;

○ c. rod = mos;

○ d. rod = rat;

Given the following code:

class B {
 int m = 7;
}

class D extends B {
 int m = 9;
}

public class TestBaseDerived {
 public static void main(String[] args) {
 B b = new B();
 D d = new D();
 B bd = new D();
 System.out.printf("%d %d %d", b.m, d.m, bd.m);
 }
}

What will be the output on executing the above code?

◉ a. 7 9 7

○ b. 9 7 9

○ c. 7 9 9

○ d. 9 9 7

Question **34**

Complete

Marked out of
1.00

☐ ▽ Flag
question

Given the following:

1. class Animal {
2.   String name = "No name";
3.   public Animal(String nm) { name = nm; }
4. }
5.
6. class DomesticAnimal extends Animal {
7.   String animalFamily = "nofamily";
8.   public DomesticAnimal(String family) { animalFamily = family; }
9. }
10.
11. public class AnimalTest {
12.   public static void main(String[] args) {
13.   DomesticAnimal da = new DomesticAnimal("cat");
14.   System.out.println(da.animalFamily);
15.   }
16. }

What is the result ?

- ○ a. nofamily
- ◉ b. Compilation fails due to an error in line 8.
- ○ c. An exception is thrown at runtime.
- ○ d. cat

Question **35**

Complete

Marked out of
1.00

☐ ▽ Flag
question

Which statement is true?

- ○ a. If neither super() nor this() is declared as the first statement in the body of a constructor, then this() will implicitly be inserted as the first statement.
- ◉ b. If both a subclass and its superclass do not have any declared constructors, the implicit default constructor of the subclass will call super() when run.
- ○ c. If super() is the first statement in the body of a constructor, then this() can be declared as the second statement.
- ○ d. A super() or this() call must always be provided explicitly as the first statement in the body of a constructor.