WEEK 8: POLYMORPHISM, ABSTRACT CLASSES, FINAL OBJECT

Given:

```
1. public class TestOverload {
2.
3.    public void process() {
4.    }
5.
6.    public String process() {
7.        return "hello";
8.    }
9.
10.   public float process(int x) {
11.       return 67.5f;
12.   }
13.}
```

What is the result?

- a. An exception is thrown at runtime.
- b. Compilation fails because of an error in line 10.
- c. Compilation succeeds and no runtime errors with class TestOverload occur.
- d. Compilation fails because of an error in line 6.

What would be the result of attempting to compile and executing the following code?

```
// Filename: MyClass.java
public class MyClass {
 public static void main(String[] args) {
  C c = new C();
  System.out.println(c.max(13, 29));
 }
}

class A {
 int max(int x, int y) {
  if (x > y)
   return x;
  else
   return y;
 }
}

class B extends A {
 int max(int x, int y) {
  return super.max(y, x) - 10;
 }
}
class C extends B {
 int max(int x, int y) {
  return super.max(x + 10, y + 10);
 }
}
```

a. code will compile without errors and will print 29 when run.

b. The code will fail to compile because a call to a max() method is ambiguous.

c. The code will fail to compile because the max() method in B passes the arguments in the call super.max(y, x) in the wrong order.

d. code will compile without errors and will print 39 when run.

Given the following:

```
1. class ParentClass {
2.     public int doStuff(int x) {
3.         return x * 2;
4.     }
5. }
6.
7. public class ChildClass extends ParentClass {
8.     public static void main(String [] args ) {
9.         ChildClass cc = new ChildClass();
10.        long x = cc.doStuff(7);
11.        System.out.println("x = " + x);
12.    }
13.
14.    public long doStuff(int x) {
15.        return x * 3;
16.     }
17. }
```

What is the result?

a. x = 14

b. Compilation fails at line 2.

c. Compilation fails at line 14.

d. x = 21

Here is a situation:

Birthday happy;

happy = new AdultBirthday("Joe", 39);

happy.greeting();

Which greeting() method is run ?

a. The one closest in the source code to the happy.greeting() statement.

b. The one defined for Birthday because that is the type of the variable happy.

c. The assignment statement where the AdultBirthday object is assigned to happy variable is an error.

d. The one defined for AdultBirthday because that is the type of the object referred to by happy.

What will be the result of attempting to compile and run the following program?

```
public class Polymorphism2 {
 public static void main(String[] args) {
  A ref1 = new C();
  B ref2 = (B) ref1;
  System.out.println(ref2.g());
 }
}

class A {
 private int f() {
  return 0;
 }

 public int g() {
  return 3;
 }
}

class B extends A {
 private int f() {
  return 1;
```

```
 }

 public int g() {
  return f();
 }
}

class C extends B {
 public int f() {
  return 2;
 }
}
```

- a. The program will compile without error and print 0 when run.
- b. The program will compile without error and print 3 when run.
- c. The program will compile without error and print 1 when run.
- d. The program will compile without error and print 2 when run.

If a class inheriting an abstract class does not define all of its function then it will be known as?

a. None of the mentioned
b. A simple class
c. Abstract
d. Static class

Which statement is true?

a. Private methods of a superclass cannot be overridden in subclasses.
b. A subclass can override any method present in a superclass.
c. An overriding method can declare that it throws more exceptions than the method it is overriding.
d. The parameter list of an overriding method must be a subset of the parameter list of the method that it is overriding.

Can an abstract class define both abstract methods and non-abstract methods?

a. No-it must have all one or the other.
b. Yes-the child classes inherit both.
c. Yes-but the child classes do not inherit the abstract methods.
d. No-it must have all abstract methods.

What is the process of defining a method in subclass having same name & type signature as a method in its superclass?

a. Method hiding
b. None of the mentioned
c. Method overloading
d. Method overriding

Given the following:

```
class A {
 public void baz() {
  System.out.println("A");
 }
}

public class B extends A {
 public static void main(String[] args) {
  A a = new B();
  a.baz();
 }
```

Given the following:

```
1. class Over {
2.     int doStuff(int a, float B. {
3.         return 7;
4.     }
5. }
6.
7. class Over2 extends Over {
8.     // insert code here
9. }
```

Which method, if inserted at line 8, will not compile?

- a. private int doStuff(int x, float y) {return 4; }
- b. public int doStuff(int x, float y) { return 4; }
- c. protected int doStuff(int x, float y) {return 4; }
- d. private int doStuff(int x, double y) { return 4; }

Which of these is supported by method overriding in Java?

- a. Polymorphism
- b. Abstraction
- c. Encapsulation
- d. None of the mentioned

Which one of the following statements is true?

- a. An abstract class can declare non-abstract methods.
- b. An abstract class can be instantiated.
- c. An abstract class is implicitly final.
- d. An abstract class can not extend a concrete class.

Which of these packages contains abstract keyword?

- a. java.util
- b. java.lang
- c. java.system
- d. java.io

What is an abstract class?

- a. An abstract class is any parent class with more than one child class.
- b. An abstract class is a class which cannot be instantiated.
- c. An abstract class is another name for "base class."
- d. An abstract class is one without any child classes.

Which declaration prevents creating a subclass of a top level class?

- a. final public class Javacg{}
- b. final abstract class Javacg{}
- c. abstract public class Javacg{}
- d. private class Javacg{}

What is an abstract method?

- a. An abstract method is a method which cannot be inherited.
- b. An abstract method is any method in an abstract class.
- c. An abstract method is a method in the child class that overrides a parent method.
- d. An abstract method is one without a body that is declared with the reserved word abstract

Which one of the following statement is false?

- a. Aggregation defines a has-a relationship between a superclass and its subclasses.
- b. Inheritance defines a is-a relationship between a superclass and its subclasses.
- c. A subclass must override all the methods of the superclass.
- d. It is possible for a subclass to define a method with the same name and parameters as a method defined by the superclass.

Which statement is true?

- a. Methods with default access in a superclass cannot be overridden in subclasses.
- b. Protected methods of a superclass cannot be overridden in subclasses.
- c. Private methods of a superclass cannot be overridden in subclasses.
- d. Public methods of a superclass cannot be overridden in subclasses.

Here is an abstract method defined in the parent:

public abstract int sumUp (int[] arr);

Which of the following is required in a non-abstract child?

- a. public int sumUp (long[] arr) { . . . }
- b. public double sumUp (int[] arr) { . . . }
- c. public int sumUp (int[] arr) { . . . }
- d. public abstract int sumUp (int[] arr) { . . . }

```
public class MySub extends MySuper {
 int calculate(int i, int j) {
  return 2 * i * j;
 }

 public static void main(String[] args) {
  MySuper sup = new MySub();
  int k = sup.calculate(2, 5);
  System.out.println(k);
 }
}
```

What is the result?

- a. 10
- b. An exception is thrown at runtime
- c. Compilation error
- d. 20

Which statement is true?

a. A final class can be abstract.

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

c. All the members of the superclass are inherited by the subclass.

d. A class in which all the members are declared private, cannot be declared public.

What is the output of this program?

```
abstract class A
{
    int i;
    abstract void display();
}
class B extends A
{
    int j;
    void display()
    {
        System.out.println(j);
    }
```

```
class Abstract_demo
{
    public static void main(String args[])
    {
        B obj = new B();
        obj.j=2;
        obj.display();
    }
}
```

- a. Compilation error
- b. 0
- c. 2
- d. Runtime error occurs.

Given:

```
1 abstract class AbstractIt
2 {
3     abstract float getFloat();
4 }
5 public class Test1 extends AbstractIt
6 {
7      private float f1 = 1.0f;
8      private float getFloat(){ return f1;}
9
10     public static void main(String[] args)
11     {
12     }
13 }
```

- a. Compilation error at line no 5
- b. Compilation succeeds
- c. Runtime error at line 8
- d. Compilation error at line no 8

Given:

```
1 abstract class AbstractIt
2 {
3     abstract float getFloat();
4 }
5 public class Test1 extends AbstractIt
6 {
7      private float f1 = 1.0f;
8      private float getFloat(){ return f1;}
9
10     public static void main(String[] args)
11     {
12     }
13 }
```

- a. Compilation error at line no 5
- b. Compilation succeeds
- c. Runtime error at line 8
- d. Compilation error at line no 8

Given the following:

```
class Foo {
    String doStuff(int x) { return "hello"; }
}
```

Which method would not be legal in a subclass of Foo?

- a. `protected String doStuff(int x) { return "Hello"; }`
- b. `String doStuff(int x) { return "hello"; }`
- c. `public String doStuff(int x) { return "Hello"; }`
- d. `int doStuff(int x) { return 42; }`

Which of these keywords can be used to prevent Method overriding?

- a. constant
- b. final
- c. static
- d. protected

Which statement is true?

- a. A subclass must define all the methods from the superclass.
- b. It is possible for two classes to be the superclass of each other.
- c. Aggregation defines a is-a relationship between a superclass and its subclasses.
- d. It is possible for a subclass to define a method with the same name and parameters as a method defined by the superclass.

Given the following classes and declarations, which statement is true?

```
// Classes
class Foo {
    private int i;
    private void f() { /* ... */ }
    public void g() { /* ... */ }
}
class Bar extends Foo {
    public int j;
    public void g() { /* ... */ }
}
// Declarations:
// ...
    Foo a = new Foo();
    Bar b = new Bar();
// ...
```

a. The statement a.g(); is legal.

b. The statement b.i = 3; is legal

c. The statement a.j = 5; is legal.

d. The statement b.f(); is legal.

A class Car and its subclass Yugo both have a method run() which was written by the programmer as part of the class definition. If junker refers to an object of type Yugo, what will the following code do?

junker.run();

a. The compiler will complain that run() has been defined twice.

b. The run() method defined in Yugo will be called.

c. The run() method defined in Car will be called.

d. Overloading will be used to pick which run() is called.

What is the output of this program?

```
class A
{
   int i;
   public void display()
   {
      System.out.println(i);
   }
}
class B extends A
{
```

What is the output of this program?

```
class A
{
    int i;
    public void display()
    {
        System.out.println(i);
    }
}
class B extends A
{
    int j;
    public void display()
    {
        System.out.println(j);
    }
}
class Dynamic_dispatch
{
    public static void main(String args[])
    {
        B obj2 = new B();
```

```
        B obj2 = new B();
        obj2.i = 1;
        obj2.j = 2;
        A r;
        r = obj2;
        r.display();
    }
}
```

- a. 4
- b. 2
- c. 3
- d. 1

Given:

```
1. abstract class AbstractClass {
2.     void setup() { }
3.     abstract int execute();
4. }
5. class EC extends AbstractClass {
6.    int execute() {
7.          System.out.println("execute of EC invoked");
8.          return 0;
9.     }
10.}
11.public class TestEC {
12.    public static void main(String... args) {
13.         EC ec = new EC();
14.         ec.setup();
15.         ec.execute();
16.    }
17.}
```

What is the expected behaviour?

a. Compilation error at line 2.

b. Prints "execute of EC invoked".

c. Compilation error at line 14.

d. Runtime error occurs.

What will be the result of attempting to compile and run the following program?

```
public class Polymorphism {
 public static void main(String[] args) {
  A ref1 = new C();
  B ref2 = (B) (ref1);
  System.out.println(ref2.f());
 }
}

class A {
 int f() {
  return 0;
 }
}
```

```
class B extends A {
 int f() {
  return 1;
 }
}

class C extends B {
 int f() {
  return 2;
 }
}
```

- a. The program will fail to compile.
- b. The program will compile without error and print 1 when run.
- c. The program will compile without error and print 2 when run.
- d. The program will compile without error, but will throw a ClassCastException when run.

Given the following,

```
1. abstract class A {
2.    abstract short m1() ;
3.    short m2() { return (short) 420; }
4. }
5.
6. abstract class B extends A {
7.    // missing code?
8.    short m1() { return (short) 42; }
9. }
```

Which of the following statements is true?

- a. Class B must either make an abstract declaration of method m2() or implement method m2() to allow the code to compile.
- b. As long as line 8 exists, class A must declare method m1() in some way.
- c. It is legal, but not required, for class B to either make an abstract declaration of method m2() or implement method m2() for the code to compile.
- d. If class A was not abstract and method m1() on line 2 was implemented, the code would not compile.

In the below class, is constructor overloaded or is method overloaded?

```
public class A
{
   public A()
   {
      //-----> (1)
   }

   void A()
   {
      //-----> (2)
   }
}
```

- a. Method
- b. Both constructor and method
- c. Constructor
- d. None of the mentioned

Given:

```
abstract class Shape {
    public abstract void draw();
}
public class Circle extends Shape {
    public void draw() { }
}
```

Which one of the following statement is correct?

a.
```
Shape s = new Circle();
s.draw();
```

b.
```
Shape s = new Circle();
s->draw();
```

c.
```
Shape s = new Shape();
s.draw();
```

d.
```
Circle c = new Shape();
c.draw();
```