

## Part I. Multiple Choice Questions (2 points each):

- Which of the following represents the life-cycle of software development?
  - Analysis -> Design -> Coding -> Testing -> Operation and Maintenance \*\*\*\*\*
  - Design -> Analysis -> Coding -> Testing -> Operation and Maintenance
  - Design -> Analysis -> Coding -> Testing
  - Analysis -> Design -> Coding -> Operation and Maintenance
- Defining a class so that the implementation of the data and methods of the class are not known to the programmers who use the class is called:
  - Data Binding
  - Polymorphism
  - Encapsulation \*\*\*\*\*
  - Inheritance
- Which of the following is an incorrect identifier?
  - 3theValue \*\*\*\*\*
  - THE\_IDENTIFIER
  - a\_b\_
  - neolithic123FOUR
- In the following block of code, what is the value of theVar?

```
int theVar = // 2
              /* /* 4 + 5 */
              6 * 3
              // - 2
              ;
```

  - 18 \*\*\*\*\*
  - 9
  - 2
  - 2
- Which of the following is the proper order of promotion?
  - short -> byte -> long -> int -> float -> double
  - short -> byte -> int -> long -> float -> double
  - short -> byte -> int -> float -> double -> long
  - byte -> short -> int -> long -> float -> double \*\*\*\*\*

6. In the following block of code, what is the value of thePhrase?

```
String S1 = "anabolic regzrding vaccination";
String S2 = "itate";
String S3 = "grad";
String thePhrase = S1.substring(S1.indexOf("r"), S1.indexOf("z")) +
                  S3.substring(1) + S2;
```

- (a) "egraditate"
- (b) "regraditate" \*\*\*\*\*
- (c) "regzgraditate"
- (d) None of these

7. Which of the following is an illegal assignment expression?

- (a) float x = 3.5; \*\*\*\*\*
- (b) int x = 3;
- (c) double x = 4.66f;
- (d) long x = (int)4;

8. What is the resulting value of the following Java expression?

```
double x = (4.0f + (3.0)/(int)1.5) * (3/(int)4.0);
```

- (a) 4.125
- (b) 3.5
- (c) 0.0 \*\*\*\*\*
- (d) 4.5

9. Given the following class and usage thereof, which of the labeled lines are incorrect?

```
public class Exam1
{
    private final int aQuandry;
    public Exam1( int quandry )
    {
I:      aQuandry = quandry;
    }
}

// ... In some other class, in some method:

II:     Exam1 exam = new Exam1();
III:    exam.aQuandry = 42;
IV:     Exam1 = new Exam1( 99 );
```

- (a) I, II
- (b) III, IV
- (c) II, III, IV \*\*\*\*\*
- (d) II, III

10. What is printed by the code below?

```
public class Test
{
    private static final int value = 5;
    public static void main( String[] args )
    {
        int total, value = 4;
        total = value + value;
        total = total + someMethod( total );
        System.out.println( total );
    }
    public static int someMethod( int val )
    {
        return value;
    }
}
```

- (a) 13 \*\*\*\*\*
- (b) None of these
- (c) 16
- (d) 15

11. A mutator method is a method that:

- (a) prints to the screen the value of a data member
- (b) reads and returns the value of a data member
- (c) changes the value of a data member \*\*\*\*\*
- (d) constructs a class

12. What is the output of the following program?

```
public class Query
{
    private static String someString = "hello";
    private String name;
    public Query( String newName )
    {
        name = newName;
    }
    public static void main( String[] args )
    {
        Query query = new Query( "Gordon" );
        changeString( someString );
        changeName( query );
        System.out.println( someString + query.name );
    }
    public static void changeString( String str )
    {
        str = "Howdy";
    }
    public static void changeName( Query q )
    {
        q.name = "Lightfoot";
    }
}
```

- (a) HowdyLightfoot
- (b) helloLightfoot \*\*\*\*\*
- (c) HowdyGordon
- (d) helloGordon

13. Which of the following boolean expressions is always true?

- (a)  $10 \leq x \ \&\& \ !( x \geq 10 )$
- (b)  $y == x + y \ \&\& \ x == x + y$
- (c)  $10 \leq x \ || \ !( x \geq 10 )$  \*\*\*\*\*
- (d)  $y == x + y \ || \ x == x + y$

14. What is displayed by the following?

```
public class Quest
{
    public Quest()
    {
    }
    public void display( String goal, String days, int adj )
    {
        System.out.println( "I am on a "+adj+" quest for the "
            +goal+" in "+days+" days." );
    }
    public static void main( String[] args )
    {
        String adj = "perilous", goal = "sticky wicket";
        int days = 3;
        Quest q = new Quest();
        q.display( adj, goal, days );
    }
}
```

- (a) I am on a sticky wicket quest for the perilous in 3 days.
- (b) I am on a perilous quest for the sticky wicket in 3 days.
- (c) I am on a 3 quest for the perilous in sticky wicket days. \*\*\*\*\*
- (d) I am on a perilous quest for the 3 in sticky wicket days.

15. Assuming: a = -1, b = -2, c = -4, d = 2, e = -1. What is the output of the following code fragment?

```
if ( a < 0 )
    if ( b < 0 )
        if ( c < 0 )
            if ( !( d < 0 && e < 0 ) )
                System.out.println( "One" );
            else
                System.out.println( "Two" );
            if ( a == e )
                System.out.println( "Three" );
            else
                System.out.println( "Four" );
```

- (a) Two  
Three
- (b) Two  
Four
- (c) One  
Four
- (d) One \*\*\*\*\*  
Three

16. Which of the following if statements is equivalent to this switch statement?

```
switch( grade ) {  
    case 5:  
    case 4:  
        a = 1;  
        b = 2;  
        break;  
    case 3:  
        a = 5;  
        break;  
    default:  
        a = 2;  
        break;  
}
```

- (a) `if ( grade == 4 || grade == 5 ) { *****  
 a = 1;  
 b = 2;  
} else if ( grade == 3 ) {  
 a = 5;  
} else {  
 a = 2;  
}`
- (b) `if( grade == 4 ) {  
 a = 1;  
 b = 2;  
} else if( grade == 3 ) {  
 a = 5;  
} else {  
 a = 2;  
}`
- (c) `if ( grade == 4 && grade == 5) {  
 a = 1;  
 b = 2;  
} else if ( grade == 3) {  
 a = 5;  
} else {  
 a = 2;  
}`
- (d) `if( grade != 5 ) {  
 if( grade == 4 ) {  
 a = 1;  
 b = 2;  
 } else if( grade == 3 ) {  
 a = 5;  
 } else {  
 a = 2;  
 }  
}`

17. Given the following StudentID class:

```
class StudentID {  
  
    private String id;  
  
    public StudentID( String newid ) {  
        id = newid;  
    }  
  
    public String getID() { return id; }  
  
    public boolean equals( StudentID otherid ) {  
        return id.equals( otherid.getID() );  
    }  
}
```

What is the output of the following code:

```
StudentID s1 = new StudentID( "8675309" );  
StudentID s2 = new StudentID( "8675309" );  
  
boolean result1 = s1 == s2;  
boolean result2 = s1.equals( s2 );  
  
System.out.println( result1 );  
System.out.println( result2 );
```

- (a) true  
true
- (b) false \*\*\*\*\*  
true
- (c) true  
false
- (d) false  
false

18. Which two of the following statements are true about constructors:

- I. A constructor has no return type and is therefore a void method.
- II. A constructor has the same name as the class.
- III. A class can have more than one constructor.
- IV. Constructors are called like any other method.

- (a) III and IV
- (b) I and II
- (c) II and III \*\*\*\*\*
- (d) I and IV

19. The basic idea of \_\_\_\_\_ is that it allows the same program instruction to mean different things in different contexts.

- (a) object oriented programming
- (b) polymorphism \*\*\*\*\*
- (c) encapsulation
- (d) inheritance

20. Complete the following Java statement to allow the instance of the Scanner class to read keyboard input.

```
Scanner keyboard = new Scanner(_____);
```

- (a) System.out
- (b) System.in \*\*\*\*\*
- (c) System.keyboard
- (d) System.input

The version of your test is **A**. Please **FILL IN CIRCLE (A) for the TEST FORM field on the BUBBLE SHEET** directly under the DATE field and turn in your exam booklet and answer sheet to the stack labeled (A) in the front of the classroom. Thank you.

## Part II. Programming Questions (60 points total):

1. (15 pts) Create a class called Kitten that has three fields: String name, Person owner, and int age.

Create a constructor for Kitten that takes a String name and a Person owner for the Kitten and uses them for initialization. Have the age for a Kitten start at 0;

Implement accessor and mutator methods for both name and owner. Make the mutator for name such that whenever a name is applied to a Kitten, the actual name of the Kitten is "<Given Name> the Feline". (e.g. given "Bob", the Kitten's name becomes "Bob the Feline")

Implement only an accessor for age. Implement a method called haveBirthday that does not return anything and simply increases a Kitten's age by one.

Finally, write a method called toString that returns a string of the form: "<Kitten name> is <age> and belongs to <Owner name>" e.g. "Bob the Feline is 87 and belongs to Gregor Samsa"

The definition for Person is found below.

```
public class Person
{
    private final String name;
    public Person( String newName )
    {
        name = newName;
    }
    public String getName()
    {
        return name;
    }
}
```

This page is left blank intentionally.

```
public class Kitten
{
    private String name;
    private Person owner;
    private int age;

    public Kitten( String name, Person owner )
    {
        setName( name );
        setOwner( owner );
        age = 0;
    }

    public void setName( String newName )
    {
        name = newName + " the Feline";
    }
    public String getName()
    {
        return name;
    }

    public void setOwner( Person newOwner )
    {
        owner = newOwner;
    }
    public Person getOwner()
    {
        return owner;
    }

    public void haveBirthday()
    {
        ++age;
    }
    public int getAge()
    {
        return age;
    }

    public String toString()
    {
        return name+" is "+age+" and belongs to "+owner.getName();
    }
}
```

2. (15 pts) Implement a Bicycle class which has the following three methods:

```
public void increaseSpeed();  
public void decreaseSpeed();  
public boolean isMoving();
```

Within the Bicycle class, you must keep track of the Bicycle's state: moving or not moving. You must also keep track of the Bicycle's current speed. Whenever the Bicycle has a positive current speed, the state should be moving. Whenever the Bicycle has a current speed of 0 the state must be not moving. You must use a boolean variable to maintain the Bicycle's state. The methods `increaseSpeed()` and `decreaseSpeed()` always increment and decrement [respectively] the current speed by 1. If `increaseSpeed()` is called on a Bicycle which is not currently moving, the Bicycle should be set to moving, and the current speed should be increased by 1. If `decreaseSpeed()` is called on a Bicycle which is moving, the current speed should be decremented by one. If the current speed is ever decreased to 0, the Bicycle's state should change from moving to not moving.

The method `isMoving()` should return the status of the Bicycle.

Your Bicycle class should also provide two constructors. One constructor takes no arguments and the other takes an integer representing the initial speed of the Bicycle. The default constructor should create a Bicycle which is not currently moving and has a current speed of 0. The second constructor should set the current speed to the passed initial speed ONLY if the speed is positive. It should also set the Bicycle's state to moving. If the initial speed given is negative or 0, the current speed should be set to 0 and the Bicycle's state should be not moving.

This page is left blank intentionally.

```
public class Bicycle {

    private boolean isMoving;
    private int currentSpeed;

    public Bicycle() {
        isMoving = false;
        currentSpeed = 0;
    }

    public Bicycle( int speed ) {
        if( speed > 0 ) {
            isMoving = true;
            currentSpeed = speed;
        } else {
            isMoving = false;
            currentSpeed = 0;
        }
    }

    public void increaseSpeed() {
        if( isMoving ) {
            currentSpeed++;
        } else {
            isMoving = true;
            currentSpeed++;
        }
    }

    public void decreaseSpeed() {
        if( isMoving ) {
            currentSpeed--;
        }

        if( currentSpeed == 0 ) {
            isMoving = false;
        }
    }
}
```

3. (15 pts) Part I: In mathematics, a polynomial equation of the second degree is commonly known as a quadratic equation. This equation can be generalized to the following:

$$ax^2 + bx + c = 0$$

We also know that there is a very simple formula to solve this equation. Recall this formula as the following:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Write a method, solveQuadratic, which takes in a, b, and c and solves the appropriate quadratic equation. Your method should return a Pair object, as defined by the class below. This object simply holds two doubles, in this case, the two doubles are the solutions to the quadratic. If the quadratic equation has no real roots (i.e, the discriminant under the square root is negative), then return a Pair object where both numbers are Double.NaN (standing for Not a Number).

```
public class Pair
{
    private double x1, x2;

    public Pair()
    {
        x1 = Double.NaN;
        x2 = Double.NaN;
    }
    public Pair(double newX1, double newX2)
    {
        x1 = newX1;
        x2 = newX2;
    }
    public void setPair(double newX1, double newX2)
    {
        x1 = newX1;
        x2 = newX2;
    }
    public double getX() { return x1; }
    public double getY() { return x2; }
}
```

```
public Pair solveQuadratic(double a, double b, double c)
{
    // write your code here

    double discriminant = b*b - 4*a*c;
    if (discriminant < 0)
        return new Pair();
    return new Pair((-b+Math.sqrt(discriminant))/(2*a),
                    (-b-Math.sqrt(discriminant))/(2*a));
}
```

(15 pts) Part II: A less known equation is a unique quartic equation called the biquadratic equation. This is a fourth order equation of the form:

$$ax^4 + bx^2 + c = 0$$

To solve a biquadratic equation, we can observe that this type of equation can be made to the form of a quadratic by substituting  $z = x^2$ . This results in the following equation:

$$az^2 + bz + c = 0$$

Using the quadratic equation, we can get a pair of solutions ( $z_1, z_2$ ). To get the four solutions to the biquadratic equation, substitute back in for  $x$ :

$$x_1 = \sqrt{z_1} \quad x_2 = -\sqrt{z_1} \quad x_3 = \sqrt{z_2} \quad x_4 = -\sqrt{z_2}$$

Using the quadratic equation solver from PART I (you may assume it is written correctly if you are unsure of your solution), write a biquadratic equation solver method, `solveBiquadratic`. Your method should call the `solveQuadratic` method to solve the quadratic equation defined by the above substitution and return a `Quad` object, which simply holds four doubles. If a particular solution to the quadratic equation is not real (that is, the `Pair` object returned from `solveQuadratic` contains a `Double.NaN` value), then its associated pair of biquadratic solutions is also not real and should be set to `Double.NaN` as well.

```
public class Quad
{
    private double x1, x2, x3, x4;

    public Quad()
    {
        x1 = Double.NaN;
        x2 = Double.NaN;
        x3 = Double.NaN;
        x4 = Double.NaN;
    }
    public Quad(double newX1, double newX2, double newX3, double newX4)
    {
        x1 = newX1;
        x2 = newX2;
        x3 = newX3;
        x4 = newX4;
    }
    public void setQuad(double newX1, double newX2, double newX3, double newX4)
    {
        x1 = newX1;
        x2 = newX2;
        x3 = newX3;
        x4 = newX4;
    }
    public double getX1() { return x1; }
    public double getX2() { return x2; }
    public double getX3() { return x3; }
    public double getX4() { return x4; }
}
```

```
public Quad solveBiquadratic(double a, double b, double c)
{
    // write your code here
}
```