Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

1. Consider the following code segment.

```
String oldStr = "ABCDEF";
String newStr = oldStr.substring(1, 3) + oldStr.substring(4);
System.out.println(newStr);
```

What is printed as a result of executing the code segment?

(A) ABCD

(B) BCDE

(C) BCEF

(D) BCDEF

(E) ABCDEF

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

2.   Consider the following class.

```
public class SomeMethods

{
public void one(int first)

{ / * implementation not shown * / }

public void one(int first, int second)

{ / * implementation not shown * / }

public void one(int first, String second)

{ / * implementation not shown * / }

}
```

Which of the following methods can be added to the SomeMethods class without causing a compile-time error?

   I. public void one(int value)
       { / * implementation not shown * / }

  II. public void one (String first, int second)

       { / * implementation not shown * / }

 III. public void one (int first, int second, int third)

       { / * implementation not shown * / }

(A)   I only
(B)   I and II only
(C)   I and III only
(D)   II and III only
(E)   I, II, and III

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

3.   A pair of number cubes is used in a game of chance. Each number cube has six sides, numbered from 1 to 6, inclusive, and there is an equal probability for each of the numbers to appear on the top side (indicating the cube's value) when the number cube is rolled. The following incomplete statement appears in a program that computes the sum of the values produced by rolling two number cubes.

int sum = / * *missing code* * / ;

Which of the following replacements for /* *missing code* */ would best simulate the value produced as a result of rolling two number cubes?

(A)   2 * (int) (Math.random() * 6)

(B)   2 * (int) (Math.random() * 7)

(C)   (int) (Math.random() * 6) + (int) (Math.random() * 6)

(D)   (int) (Math.random() * 13)

(E)   2 + (int) (Math.random() * 6) + (int) (Math.random() * 6)

4.   Assume that myList is an ArrayList that has been correctly constructed and populated with objects. Which of the following expressions produces a valid random index for myList?

(A)   (int) (Math.random () * myList.size ()) - 1

(B)   (int) (Math.random () * myList.size ())

(C)   (int) (Math.random () * myList.size ()) + 1

(D)   (int) (Math.random () * (myList.size () + 1))

(E)   Math.random (myList.size ())

5.   Consider the following methods, which appear in the same class.

```
public void slope(int x1, int y1, int x2, int y2)
{
    int xChange = x2 - x1;
    int yChange = y2 - y1;
    printFraction(yChange, xChange);
}

public void printFraction(int numerator, int denominator)
{
    System.out.print(numerator + "/" + denominator);
}
```

Assume that the method call slope(1, 2, 5, 10) appears in a method in the same class. What is printed as a result of the method call?

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

(A) 8/4

(B) 5/1

(C) 4/8

(D) 2/1

(E) 1/5

6.

Consider the following method, which is intended to calculate and return the expression $\sqrt{\frac{(x+y)^2}{|a-b|}}$.

```
public double calculate(double x, double y, double a, double b)
{
    return /* missing code */;
}
```

Which of the following can replace /* *missing code* */ so that the method works as intended?

(A) `Math.sqrt(x ^ 2, y ^ 2, a - b)`

(B) `Math.sqrt((x + y) ^ 2) / Math.abs(a, b)`

(C) `Math.sqrt((x + y) ^ 2 / Math.abs(a - b))`

(D) `Math.sqrt(Math.pow(x + y, 2) / Math.abs(a, b))`

(E) `Math.sqrt(Math.pow(x + y, 2) / Math.abs(a - b))`

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

7. Consider the following class declaration

```
public class SomeClass
{
  private int num;

  public SomeClass(int n)
  {
    num = n;
  }

  public void increment(int more)
  {
    num = num + more;
  }

  public int getNum()
  {
    return num;
  }
}
```

The following code segment appears in another class.

```
SomeClass one = new SomeClass(100);
```

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

SomeClass two = new SomeClass(100);

SomeClass three = one;

one.increment(200);

System.out.println(one.getNum() + " " + two.getNum() + " " +

three.getNum());

What is printed as a result of executing the code segment?
- (A) 100 100 100
- (B) 300 100 100
- (C) 300 100 300
- (D) 300 300 100
- (E) 300 300 300

8. Consider the following method.

```
public double myMethod(int a, boolean b)
{   /* implementation not shown */   }
```

Which of the following lines of code, if located in a method in the same class as `myMethod`, will compile without error?

- (A) `int result = myMethod(2, false);`
- (B) `int result = myMethod(2.5, true);`
- (C) `double result = myMethod(0, false);`
- (D) `double result = myMethod(true, 10);`
- (E) `double result = myMethod(2.5, true);`

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

9. Consider the following class declaration.

```
public class Sample
{
    private int a;
    private double b;

    public Sample(int x, double y)
    {
        a = x;
        b = y;
    }

    // No other constructors
}
```

The following method appears in a class other than `Sample`.

```
public static void test()
{
    Sample object = new /* missing constructor call */ ;
}
```

Which of the following could be used to replace /* *missing constructor call* */ so that the method will compile without error?

(A) `Sample()`

(B) `Sample(int x = 10, double y = 6.2)`

(C) `Sample(int x, double y)`

(D) `Sample(10, 6.2)`

(E) `Sample(6.2, 10)`

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

10. Consider the following method.

```
/** Precondition: Strings one and two have the same length. */
public static String combine(String one, String two)
{
    String res = "";
    for (int k = 0; k < one.length(); k++)
    {
        if (one.substring(k, k + 1).equals(two.substring(k, k + 1)))
        {
            res += one.substring(k, k + 1);
        }
        else
        {
            res += "0";
        }
    }
    return res;
}
```

What is returned as a result of the call `combine("10110", "01100")` ?

(A) `"00000"`

(B) `"00100"`

(C) `"00101"`

(D) `"10110"`

(E) `"11011"`

11. The following statement assigns an integer value to `x`.

```
int x = (int)(Math.random() * 5) + 10;
```

Consider the statement that would result if the positions of `5` and `10` were swapped in the preceding statement and the resulting integer were assigned to `y`.

```
int y = (int)(Math.random() * 10) + 5;
```

Which of the following are true statements about how the possible values assigned to `y` differ from the possible values assigned to `x` ?

    I.   There are more possible values of `x` than there are possible values of `y`.
   II. There are more possible values of `y` than there are possible values of `x`.
  III. The value assigned to `x` can sometimes be the same as the value assigned to `y`.

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

   (A) I only

   (B) II only

   (C) III only

   (D) I and III

   (E) II and III

12. Consider the following code segment.

```
String str = "CompSci";
System.out.println(str.substring(0, 3));
int num = str.length();
```

What is the value of `num` when the code segment is executed?

   (A) 3

   (B) 4

   (C) 5

   (D) 6

   (E) 7

13. Consider the following code segment.

```
String temp = "comp";
System.out.print(temp.substring(0) + " " +
            temp.substring(1) + " " +
            temp.substring(2) + " " +
            temp.substring(3));
```

What is printed when the code segment is executed?

   (A) comp

   (B) c o m p

   (C) comp com co c

   (D) comp omp mp p

   (E) comp comp comp comp

14. Consider the following code segment.

```
String str = "0";
str += str + 0 + 8;
System.out.println(str);
```

What is printed as a result of executing the code segment?

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

(A) 8

(B) 08

(C) 008

(D) 0008

(E) Nothing is printed, because numerical values cannot be added to a `String` object.

15. Consider the following code segment.

```
int one = 1;
int two = 2;
String zee = "Z";
System.out.println(one + two + zee);
```

What is printed as a result of executing the code segment?

(A) 12Z

(B) 3Z

(C) 12zee

(D) 3zee

(E) onetwozee

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

16. Consider the following class declaration.

```
public class Person
{
  private String myName;
  private int myYearOfBirth;

  public Person(String name, int yearOfBirth)
  {
    myName = name;
    myYearOfBirth = yearOfBirth;
  }

  public String getName()
  {   return myName;   }

  public void setName(String name)
  {   myName = name;   }

  // There may be instance variables, constructors, and methods that are not shown.
}
```

Assume that the following declaration has been made.
Person student = new Person ("Thomas", 1995);
Which of the following statements is the most appropriate for changing the name of student from "Thomas" to "Tom" ?

(A) student = new Person ("Tom", 1995);

(B) student.myName = "Tom";

(C) student.getName ("Tom");

(D) student.setName ("Tom");

(E) Person.setName ("Tom");

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

17. Consider the following class declaration.

```
public class Student
{
  private String myName;
  private int myAge;

  public Student()
  {  /* implementation not shown */  }

  public Student(String name, int age)
  {  /* implementation not shown */  }

  // No other constructors
}
```

Which of the following declarations will compile without error?

   I. Student a = new Student();

  II. Student b = new Student("Juan", 15);

  III. Student c = new Student("Juan", "15");

(A) I only

(B) II only

(C) I and II only

(D) I and III only

(E) I, II, and III

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

Directions: Select the choice that best fits each statement. The following question(s) refer to the following incomplete class declaration.

```java
public class TimeRecord
{
  private int hours;
  private int minutes; // 0 <= minutes < 60
  /** Constructs a TimeRecord object.
   *   @param h  the number of hours
   *              Precondition: h >= 0
   *   @param m  the number of minutes
   *              Precondition: 0 <= m < 60
   */
  public TimeRecord(int h, int m)
  {
    hours = h;
    minutes = m;
  }

  /** @return  the number of hours
   */
  public int getHours()
  { /* implementation not shown */ }

  /** @return  the number of minutes
   *   Postcondition: 0 <= minutes < 60
   */
  public int getMinutes()
  { /* implementation not shown */ }

  /** Adds h hours and m minutes to this TimeRecord.
   *   @param h  the number of hours
   *              Precondition: h >= 0
   *   @param m  the number of minutes
   *              Precondition: m >= 0
   */
  public void advance(int h, int m)
  {
    hours = hours + h;
    minutes = minutes + m;
    /* missing code */
  }
}
// Other methods not shown
```

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

18. Consider the following declaration that appears in a class other than TimeRecord.

TimeRecord [ ] timeCards = new TimeRecord [100] ;

Assume that timeCards has been initialized with TimeRecord objects. Consider the following code segment that is intended to compute the total of all the times stored in timeCards.

```
TimeRecord total = new TimeRecord(0,0);
for (int k = 0; k < timeCards.length; k++)
{
   /* missing expression */ ;
}
```

Which of the following can be used to replace / * *missing expression* * / so that the code segment will work as intended?

(A) timeCards [k] .advance ( )

(B) total += timeCards [k] .advance ( )

(C)
```
total.advance(timeCards[k].hours,
                timeCards[k].minutes)
```

(D)
```
total.advance(timeCards[k].getHours(),
                timeCards[k].getMinutes())
```

(E)
```
timeCards[k].advance(timeCards[k].getHours(),
                    timeCards[k].getMinutes())
```

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

19. Consider the following instance variable and method.

```
private List<String> animals;

public void manipulate()
{
  for (int k = animals.size() - 1; k > 0; k--)
  {
    if (animals.get(k).substring(0, 1).equals("b"))
    {
      animals.add(animals.size() - k, animals.remove(k));
    }
  }
}
```

Assume that animals has been instantiated and initialized with the following contents.

```
["bear", "zebra", "bass", "cat", "koala", "baboon"]
```

What will the contents of animals be as a result of calling manipulate?

(A) ["baboon", "zebra", "bass", "cat", "bear", "koala"]

(B) ["bear", "zebra", "bass", "cat", "koala", "baboon"]

(C) ["baboon", "bear", "zebra", "bass", "cat", "koala"]

(D) ["bear", "baboon", "zebra", "bass", "cat", "koala"]

(E) ["zebra", "cat", "koala", "baboon", "bass", "bear"]

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

20. Consider the following method.

public int pick(boolean test, int x, int y)

{

if (test)

return x;

else

return y;

}

What value is returned by the following method call?

pick(false, pick(true, 0, 1), pick(true, 6, 7))

(A) 0
(B) 1
(C) 3
(D) 6
(E) 7

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

The following questions refer to the code from the GridWorld case study. A copy of the code is provided below.

Appendix B — Testable API

info.gridworld.grid.Location class (implements Comparable)

public Location(int r, int c)

      constructs a location with given row and column coordinates

public int getRow()

      returns the row of this location

public int getCol()

      returns the column of this location

public Location getAdjacentLocation(int direction)

      returns the adjacent location in the direction that is closest to direction

public int getDirectionToward(Location target)

      returns the closest compass direction from this location toward target

public boolean equals(Object other)

      returns true if other is a Location with the same row and column as this location; false otherwise

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

public int hashCode()

> returns a hash code for this location

public int compareTo(Object other)

> returns a negative integer if this location is less than other, zero if the two locations are equal, or a positive integer if this location is greater than other. Locations are ordered in row-major order.

> **Precondition:** other is a Location object.

public String toString()

> returns a string with the row and column of this location, in the format (row, col)

Compass directions:

public static final int NORTH = 0;

public static final int EAST = 90;

public static final int SOUTH = 180;

public static final int WEST = 270;

public static final int NORTHEAST = 45;

public static final int SOUTHEAST = 135;

public static final int SOUTHWEST = 225;

public static final int NORTHWEST = 315;

Turn angles:

public static final int LEFT = -90;

public static final int RIGHT = 90;

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

public static final int HALF_LEFT = -45;

public static final int HALF_RIGHT = 45;

public static final int FULL_CIRCLE = 360;

public static final int HALF_CIRCLE = 180;

public static final int AHEAD = 0;

info.gridworld.grid.Grid<E> interface

int getNumRows()

> returns the number of rows, or -1 if this grid is unbounded

int getNumCols()

> returns the number of columns, or -1 if this grid is unbounded

boolean isValid(Location loc)

> returns true if loc is valid in this grid, false otherwise

> **Precondition:** loc is not null

E put(Location loc, E obj)

> puts obj at location loc in this grid and returns the object previously at that location (or null if the location was previously unoccupied).

> **Precondition:** (1) loc is valid in this grid (2) obj is not null

E remove(Location loc)

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

removes the object at location loc from this grid and returns the object that was removed (or null if the location is unoccupied)

Precondition: loc is valid in this grid

E get(Location loc)

returns the object at location loc (or null if the location is unoccupied)

Precondition: loc is valid in this grid

ArrayList<Location> getOccupiedLocations()

returns an array list of all occupied locations in this grid

ArrayList<Location> getValidAdjacentLocations(Location loc)

returns an array list of the valid locations adjacent to loc in this grid

Precondition: loc is valid in this grid

ArrayList<Location> getEmptyAdjacentLocations(Location loc)

returns an array list of the valid empty locations adjacent to loc in this grid

Precondition: loc is valid in this grid

ArrayList<Location> getOccupiedAdjacentLocations(Location loc)

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

returns an array list of the valid occupied locations adjacent to loc in this grid

Precondition: loc is valid in this grid

ArrayList<E> getNeighbors(Location loc)

returns an array list of the objects in the occupied locations adjacent to loc in this grid

Precondition: loc is valid in this grid

info.gridworld.actor.Actor class

public Actor()

constructs a blue actor that is facing north

public Color getColor()

returns the color of this actor

public void setColor(Color newColor)

sets the color of this actor to newColor

public int getDirection()

returns the direction of this actor, an angle between 0 and 359 degrees

public void setDirection(int newDirection)

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

sets the direction of this actor to the angle between 0 and 359 degrees that is equivalent to newDirection

public Grid<Actor> getGrid()

returns the grid of this actor, or null if this actor is not contained in a grid

public Location getLocation()

returns the location of this actor, or null if this actor is not contained in a grid

public void putSelfInGrid(Grid<Actor> gr, Location loc)

puts this actor into location loc of grid gr. If there is another actor at loc, it is removed.

Precondition: (1) This actor is not contained in a grid (2) loc is valid in gr

public void removeSelfFromGrid()

removes this actor from its grid.

Precondition: this actor is contained in a grid

public void moveTo(Location newLocation)

moves this actor to newLocation. If there is another actor at newLocation, it is removed.

Precondition: (1) This actor is contained in a grid (2) newLocation is valid in the grid of this actor

public void act()

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

reverses the direction of this actor. Override this method in subclasses of Actor to define types of actors with different behavior

public String toString()

returns a string with the location, direction, and color of this actor

info.gridworld.actor.Rock class (extends Actor)

public Rock()

constructs a black rock

public Rock(Color rockColor)

constructs a rock with color rockColor

public void act()

overrides the act method in the Actor class to do nothing

info.gridworld.actor.Flower class (extends Actor)

public Flower()

constructs a pink flower

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

public Flower(Color initialColor)

   constructs a flower with color initialColor

public void act()

  causes the color of this flower to darken

21. Consider the following method that is intended to move the parameter anActor to a different grid that is referred to by the parameter newGrid. The location of anActor in newGrid should be the same as the location that anActor had occupied in its original grid.

   /** Moves anActor to newGrid in the same location it occupied in its original grid.

   * @param anActor the actor to be moved

   * @param newGrid the grid in which the actor is to be placed

   */

   public void moveActorToNewGrid(Actor anActor, Grid<Actor> newGrid)

   {

   Grid<Actor> oldGrid = anActor.getGrid();

   Location loc = anActor.getLocation();

   /* missing code */

   }

   Which of the following can be used to replace /* *missing code* */ so that moveActorToNewGrid will work as intended?

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

(A) anActor.putSelfInGrid(newGrid, loc);
anActor.removeSelfFromGrid();

(B) oldGrid.remove(loc);
anActor.putSelfInGrid(newGrid, loc);

(C) anActor.removeSelfFromGrid();
anActor.putSelfInGrid(newGrid, loc);

(D) oldGrid.remove(loc);
newGrid.put(anActor, loc);

(E) newGrid.put(anActor, loc);
oldGrid.remove(loc);

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

The following questions refer to the code in the GridWorld case study. A copy of the code is provided below.

Consider the design of a Grasshopper class that extends Bug. When asked to move, a Grasshopper moves to a randomly chosen empty adjacent location that is within the grid. If there is no empty adjacent location that is within the grid, the Grasshopper does not move, but turns 45 degrees to the right without changing its location.

Appendix B — Testable API

info.gridworld.grid.Location class (implements Comparable)

public Location(int r, int c)

　　constructs a location with given row and column coordinates

public int getRow()

　　returns the row of this location

public int getCol()

　　returns the column of this location

public Location getAdjacentLocation(int direction)

　　returns the adjacent location in the direction that is closest to direction

public int getDirectionToward(Location target)

　　returns the closest compass direction from this location toward target

public boolean equals(Object other)

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

returns true if other is a Location with the same row and column as this location; false otherwise

public int hashCode()

returns a hash code for this location

public int compareTo(Object other)

returns a negative integer if this location is less than other, zero if the two locations are equal, or a positive integer if this location is greater than other. Locations are ordered in row-major order.

Precondition: other is a Location object.

public String toString()

returns a string with the row and column of this location, in the format (row, col)

Compass directions:

public static final int NORTH = 0;

public static final int EAST = 90;

public static final int SOUTH = 180;

public static final int WEST = 270;

public static final int NORTHEAST = 45;

public static final int SOUTHEAST = 135;

public static final int SOUTHWEST = 225;

public static final int NORTHWEST = 315;

Turn angles:

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

public static final int LEFT = -90;

public static final int RIGHT = 90;

public static final int HALF_LEFT = -45;

public static final int HALF_RIGHT = 45;

public static final int FULL_CIRCLE = 360;

public static final int HALF_CIRCLE = 180;

public static final int AHEAD = 0;

info.gridworld.grid.Grid<E> interface

int getNumRows()

> returns the number of rows, or -1 if this grid is unbounded

int getNumCols()

> returns the number of columns, or -1 if this grid is unbounded

boolean isValid(Location loc)

> returns true if loc is valid in this grid, false otherwise

> **Precondition:** loc is not null

E put(Location loc, E obj)

> puts obj at location loc in this grid and returns the object previously at that location (or null if the

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

location was previously unoccupied).

Precondition: (1) loc is valid in this grid (2) obj is not null

E remove(Location loc)

removes the object at location loc from this grid and returns the object that was removed (or null if the location is unoccupied)

Precondition: loc is valid in this grid

E get(Location loc)

returns the object at location loc (or null if the location is unoccupied)

Precondition: loc is valid in this grid

ArrayList<Location> getOccupiedLocations()

returns an array list of all occupied locations in this grid

ArrayList<Location> getValidAdjacentLocations(Location loc)

returns an array list of the valid locations adjacent to loc in this grid

Precondition: loc is valid in this grid

ArrayList<Location> getEmptyAdjacentLocations(Location loc)

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

returns an array list of the valid empty locations adjacent to loc in this grid

Precondition: loc is valid in this grid

ArrayList<Location> getOccupiedAdjacentLocations(Location loc)

returns an array list of the valid occupied locations adjacent to loc in this grid

Precondition: loc is valid in this grid

ArrayList<E> getNeighbors(Location loc)

returns an array list of the objects in the occupied locations adjacent to loc in this grid

Precondition: loc is valid in this grid

info.gridworld.actor.Actor class

public Actor()

constructs a blue actor that is facing north

public Color getColor()

returns the color of this actor

public void setColor(Color newColor)

sets the color of this actor to newColor

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

public int getDirection()

   returns the direction of this actor, an angle between 0 and 359 degrees

public void setDirection(int newDirection)

   sets the direction of this actor to the angle between 0 and 359 degrees that is equivalent to newDirection

public Grid<Actor> getGrid()

   returns the grid of this actor, or null if this actor is not contained in a grid

public Location getLocation()

   returns the location of this actor, or null if this actor is not contained in a grid

public void putSelfInGrid(Grid<Actor> gr, Location loc)

   puts this actor into location loc of grid gr. If there is another actor at loc, it is removed.

   Precondition: (1) This actor is not contained in a grid (2) loc is valid in gr

public void removeSelfFromGrid()

   removes this actor from its grid.

   Precondition: this actor is contained in a grid

public void moveTo(Location newLocation)

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

moves this actor to newLocation. If there is another actor at newLocation, it is removed.

Precondition: (1) This actor is contained in a grid (2) newLocation is valid in the grid of this actor

public void act()

reverses the direction of this actor. Override this method in subclasses of Actor to define types of actors with different behavior

public String toString()

returns a string with the location, direction, and color of this actor

info.gridworld.actor.Rock class (extends Actor)

public Rock()

constructs a black rock

public Rock(Color rockColor)

constructs a rock with color rockColor

public void act()

overrides the act method in the Actor class to do nothing

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

info.gridworld.actor.Flower class (extends Actor)

public Flower()

>	constructs a pink flower

public Flower(Color initialColor)

>	constructs a flower with color initialColor

public void act()

>	causes the color of this flower to darken

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

22. Consider the following method that is intended to return an ArrayList of all the locations in grd that contain actors facing in direction dir.

public ArrayList<Location> findLocsFacingDir(int dir, Grid<Actor> grd)

{

ArrayList<Location> desiredLocs = new ArrayList<Location>();

for (Location loc : grd.getOccupiedLocations())

{

if (/* *expression* */ == dir)

desiredLocs.add(loc);

}

return desiredLocs;

}

Which of the following can be used to replace /* *expression* */ so that findLocsFacingDir will work as intended?

(A) loc.getDirection()

(B) getDirection(loc)

(C) ((Actor) loc).getDirection()

(D) grd(loc).getDirection()

(E) grd.get(loc).getDirection()

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

23. Consider the following method.

```
public String mystery(String input)
{
  String output = "";

  for (int k = 1; k < input.length(); k = k + 2)
  {
    output += input.substring(k, k + 1);
  }

  return output;
}
```

What is returned as a result of the call mystery("computer") ?

(A) "computer"

(B) "cmue"

(C) "optr"

(D) "ompute"

(E) Nothing is returned because an IndexOutOfBoundsException is thrown.

24. Consider the following method.

```
public static String scramble(String word, int howFar)
{
  return word.substring(howFar + 1, word.length()) +
         word.substring(0, howFar);
}
```

What value is returned as a result of the call scramble("compiler", 3)?

(A) "compiler"

(B) "pilercom"

(C) "ilercom"

(D) "ilercomp"

(E) No value is returned because an IndexOutOfBoundsException will be thrown.

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

25. Consider the following method, which is intended to return true if at least one of the three strings s1, s2, or s3 contains the substring "art". Otherwise, the method should return false.

```
public static boolean containsArt(String s1, String s2, String s3)
{

  String all = s1 + s2 + s3;

  return (all.indexOf("art") != -1);

}
```

Which of the following method calls demonstrates that the method does not work as intended?

(A)   containsArt ("rattrap", "similar", "today")

(B)   containsArt ("start", "article", "Bart")

(C)   containsArt ("harm", "chortle", "crowbar")

(D)   containsArt ("matriculate", "carat", "arbitrary")

(E)   containsArt ("darkroom", "cartoon", "articulate")

26. Consider the following method, which is intended to return the element of a 2-dimensional array that is closest in value to a specified number, val.

```
/** @return  the element of 2-dimensional array  mat  whose value is closest to  val  */
public double findClosest(double[][] mat, double val)
{
  double answer = mat[0][0];
  double minDiff = Math.abs(answer - val);
  for (double[] row : mat)
  {
    for (double num : row)
    {
      if ( /* missing code */ )
      {
        answer = num;
        minDiff = Math.abs(num - val);
      }
    }
  }
  return answer;
}
```

Which of the following could be used to replace / * *missing code* * / so that findClosest will work as intended?

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

    (A)   val - row [num] < minDiff

    (B)   Math.abs (num - minDiff) < minDiff

    (C)   val - num < 0.0

    (D)   Math.abs (num - val) < minDiff

    (E)   Math.abs (row [num] - val) < minDiff

27.   Consider the following method.

```
public static boolean mystery(String str)
{
  String temp = "";

  for (int k = str.length(); k > 0; k--)
  {
    temp = temp + str.substring(k - 1, k);
  }

  return temp.equals(str);
}
```

Which of the following calls to mystery will return true?

    (A)   mystery ("no")

    (B)   mystery ("on")

    (C)   mystery ("nnoo")

    (D)   mystery ("nono")

    (E)   mystery ("noon")

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

28. Consider the following method, which is intended to count the number of times the letter `"A"` appears in the string `str`.

```
public static int countA(String str)
{
    int count = 0;

    while (str.length() > 0)
    {
        int pos = str.indexOf("A");
        if (pos >= 0)
        {
            count++;
            /* missing code */
        }
        else
        {
            return count;
        }
    }
    return count;
}
```

Which of the following should be used to replace /* *missing code* */ so that method `countA` will work as intended?

(A) `str = str.substring(0, pos);`

(B) `str = str.substring(0, pos + 1);`

(C) `str = str.substring(pos - 1);`

(D) `str = str.substring(pos);`

(E) `str = str.substring(pos + 1);`

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

29. The `Student` class has been defined to store and manipulate grades for an individual student. The following methods have been defined for the class.

```
/* Returns the sum of all of the student's grades */
public double sumOfGrades()
{  /* implementation not shown */  }
/* Returns the total number of grades the student has received */
public int numberOfGrades()
{  /* implementation not shown */  }
/* Returns the lowest grade the student has received */
public double lowestGrade()
{  /* implementation not shown */  }
```

Which of the following statements, if located in a method in the `Student` class, will determine the average of all of the student's grades except for the lowest grade and store the result in the `double` variable `newAverage` ?

(A) `newAverage = sumOfGrades() / numberOfGrades() - 1;`

(B) `newAverage = sumOfGrades() / (numberOfGrades() - 1);`

(C) `newAverage = sumOfGrades() - lowestGrade() / (numberOfGrades() - 1);`

(D) `newAverage = (sumOfGrades() - lowestGrade()) / numberOfGrades() - 1;`

(E) `newAverage = (sumOfGrades() - lowestGrade()) / (numberOfGrades() - 1);`

30. Consider the following method.

```
public void doSomething()
{
    System.out.println("Something has been done");
}
```

Each of the following statements appears in a method in the same class as `doSomething`. Which of the following statements are valid uses of the method `doSomething` ?

```
  I. doSomething();
 II. String output = doSomething();
III. System.out.println(doSomething());
```

(A) I only

(B) II only

(C) I and II only

(D) I and III only

(E) I, II, and III

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

31. Consider the following code segment.

```
double d1 = 10.0;
Double d2 = 20.0;
Double d3 = new Double(30.0);
double d4 = new Double(40.0);

System.out.println(d1 + d2 + d3.doubleValue() + d4);
```

What, if anything, is printed when the code segment is executed?

(A) `100.0`

(B) `10.050.040.0`

(C) `10.020.070.0`

(D) `10.020.030.040.0`

(E) There is no output due to a compilation error.

32. Consider the following class definition.

```
public class ExamScore
{
    private String studentId;
    private double score;
    public ExamScore(String sid, double s)
    {
        studentId = sid;
        score = s;
    }
    public double getScore()
    {
        return score;
    }
    public void bonus(int b)
    {
        score += score * b/100.0;
    }
}
```

Assume that the following code segment appears in a class other than `ExamScore`.

```
ExamScore es = new ExamScore("12345", 80.0);
es.bonus(5);
System.out.println(es.getScore());
```

What is printed as a result of executing the code segment?

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

    (A) 4.0

    (B) 5.0

    (C) 80.0

    (D) 84.0

    (E) 85.0

Consider the following method.

```
public static String[] strArrMethod(String[] arr)
{
    String[] result = new String[arr.length];
    for (int j = 0; j < arr.length; j++)
    {
        String sm = arr[j];
        for (int k = j + 1; k < arr.length; k++)
        {
            if (arr[k].length() < sm.length())
            {
        sm = arr[k]; // Line 12
            }
        }
        result[j] = sm;
    }
    return result;
}
```

33. Consider the following code segment.

```
String[] testOne = {"first", "day", "of\", "spring"};
String[] resultOne = strArrMethod(testOne);
```

What are the contents of `resultOne` when the code segment has been executed?

    (A) {"day", "first", "of\", "spring"}

    (B) {"of\", "day", "first", "spring"}

    (C) {"of\", "day", "of\", "spring"}

    (D) {"of\", "of\", "of\", "spring"}

    (E) {"spring", "first", "day", "of\"}

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

34. Consider the following code segment.

```
String[] testTwo = {"last", "day", "of\", "the", "school", "year"};
String[] resultTwo = strArrMethod(testTwo);
```

How many times is the line labeled `// Line 12` in the `strArrMethod` executed as a result of executing the code segment?

(A) 4 times

(B) 5 times

(C) 6 times

(D) 15 times

(E) 30 times

35. Consider the following method.

```
public static int what(String str, String check)
{
    int num = -1;
    int len = check.length();
    for (int k = 0; k + len <= str.length(); k++)
    {
        String a = str.substring(k, k + len);
        if (a.equals(check))
        {
            num = k;
        }
    }
    return num;
}
```

Assume that `check` occurs at least once in `str`. Which of the following best describes the value returned by the `what` method?

(A) The number of times the string `check` occurs in `str`

(B) The index of the first occurrence of `check` inside `str`

(C) The index of the last occurrence of `check` inside `str`

(D) The number of substrings in `str` with the same length as `check`

(E) The number of substrings in `str` that do not match `check`

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

36.   Consider the following methods, which appear in the same class.

```
public int function1(int i, int j)
{
     return i + j;
}

public int function2(int i, int j)
{
     return j - i;
}
```

Which of the following statements, if located in a method in the same class, will initialize the variable `x` to `11`?

(A) `int x = function2(4, 5) + function1(1, 3);`

(B) `int x = function1(4, 5) + function2(1, 3);`

(C) `int x = function1(4, 5) + function2(3, 1);`

(D) `int x = function1(3, 1) + function2(4, 5);`

(E) `int x = function2(3, 1) + function1(4, 5);`

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

37. Consider the following class declaration.

```
public class GameClass
{
    private int numPlayers;
    private boolean gameOver;

    public Game()
    {
        numPlayers = 1;
        gameOver = false;
    }

    public void addPlayer()
    {
        numPlayers++;
    }

    public void endGame()
    {
        gameOver = true;
    }
}
```

Assume that the `GameClass` object `game` has been properly declared and initialized in a method in a class other than `GameClass`. Which of the following statements are valid?

    I. `game.numPlayers++;`
   II. `game.addPlayer();`
  III. `game.gameOver();`
  IV. `game.endGame();`

(A) IV only

(B) I and III only

(C) I and IV only

(D) II and IV only

(E) II, III, and IV only

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

38. Consider the method `getHours`, which is intended to calculate the number of hours that a vehicle takes to travel between two *mile markers* on a highway if the vehicle travels at a constant speed of 60 miles per hour. A mile marker is a sign showing the number of miles along a road between some fixed location (for example, the beginning of a highway) and the current location.

    The following table shows two examples of the intended behavior of `getHours`, based on the `int` parameters `marker1` and `marker2`.

marker1	marker2	Return Value
100	220	2.0
100	70	0.5

    Consider the following implementation of `getHours`.

    ```
    public static double getHours(int marker1, int marker2)
    {
        /* missing statement */
        return hours;
    }
    ```

    Which of the following statements can replace /* *missing statement* */ so `getHours` works as intended?

    (A) `double hours = (Math.abs(marker1) - Math.abs(marker2)) / 60.0;`

    (B) `double hours = Math.abs(marker1 - marker2 / 60.0);`

    (C) `double hours = Math.abs(marker1 - marker2) / 60.0;`

    (D) `double hours = Math.abs((marker1 - marker2) / 60);`

    (E) `double hours = (double) (Math.abs(marker1 - marker2) / 60);`

39. Consider the following code segment.

    String str = "abcdef";

    for (int rep = 0; rep < str.length() - 1; rep++)

    {

      System.out.print(str.substring(rep, rep + 2));

    }

    What is printed as a result of executing this code segment?

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

    (A) abcdef

    (B) aabbccddeeff

    (C) abbccddeef

    (D) abcbcdcdedef

    (E) Nothing is printed because an IndexOutOfBoundsException is thrown.

40. Consider the following class definition.

```
public class Bird
{
    private String species;
    private String color;
    private boolean canFly;
    public Bird(String str, String col, boolean cf)
    {
        species = str;
        color = col;
        canFly = cf;
    }
}
```

Which of the following constructors, if added to the `Bird` class, will cause a compilation error?

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

(A)
```
public Bird()
{
     species = "unknown";
     color = "unknown";
     canFly = false;
}
```

(B)
```
public Bird(boolean cf)
{
     species = "unknown";
     color = "unknown";
     canFly = cf;
}
```

(C)
```
public Bird(String col, String str)
{
     species = str;
     color = col;
     canFly = false;
}
```

(D)
```
public Bird(boolean cf, String str, String col)
{
     species = str;
     color = col;
     canFly = cf;
}
```

(E)
```
public Bird(String col, String str, boolean cf)
{
     species = str;
     color = col;
     canFly = cf;
}
```

41. A student has created a `Car` class. The class contains variables to represent the following.

A `String` variable called `color` to represent the color of the car
An `int` variable called `year` to represent the year the car was made
A `String` variable called `make` to represent the manufacturer of the car
A `String` variable called `model` to represent the model of the car

The object `vehicle` will be declared as type `Car`.

Which of the following descriptions is accurate?

(A) An instance of the `vehicle` class is `Car`.

(B) An instance of the `Car` object is `vehicle`.

(C) An attribute of the `year` object is `int`.

(D) An attribute of the `vehicle` object is `color`.

(E) An attribute of the `Car` instance is `vehicle`.

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

42. Consider the code segment below.

```
int a = 1988;
int b = 1990;

String claim = " that the world's athletes " +
          "competed in Olympic Games in ";

String s = "It is " + true + claim + a +
        " but " + false + claim + b + ".";

System.out.println(s);
```

What, if anything, is printed when the code segment is executed?

(A) `It is trueclaima but falseclaimb.`

(B) `It is trueclaim1998 but falseclaim1990.`

(C) `It is true that the world's athletes competed in Olympic Games in a but false that the world's athletes competed in Olympic Games in b.`

(D) `It is true that the world's athletes competed in Olympic Games in 1988 but false that the world's athletes competed in Olympic Games in 1990.`

(E) Nothing is printed because the code segment does not compile.

43. A student has created an `OrderedPair` class to represent points on an xy-plane. The class contains the following.

An `int` variable called `x` to represent an x-coordinate.
An `int` variable called `y` to represent a y-coordinate.
A method called `printXY` that will print the values of `x` and `y`.

The object `origin` will be declared as type `OrderedPair`.

Which of the following descriptions is accurate?

(A) `origin` is an instance of the `printXY` method.

(B) `origin` is an instance of the `OrderedPair` class.

(C) `origin` is an instance of two `int` objects.

(D) `OrderedPair` is an instance of the `origin` object.

(E) `printXY` is an instance of the `OrderedPair` class.

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

44. Consider the following attempts at method overloading.

I.
```
public class Overload
{
    public int average(int x, int y)
    { /* implementation not shown */ }

    public int average(int value1, int value2)
    { /* implementation not shown */ }

    // There may be instance variables, constructors,
    // and methods that are not shown.
}
```

II.
```
public class Overload
{
    public int average(int x, int y)
    { /* implementation not shown */ }

    public int average(int x, int y, int z)
    { /* implementation not shown */ }

    // There may be instance variables, constructors
    // and methods that are not shown.
}
```

III.
```
public class Overload
{
    public int average(int x, int y)
    { /* implementation not shown */ }

    public int average(double x, double y)
    { /* implementation not shown */ }

    // There may be instance variables, constructors,
    // and methods that are not shown.
}
```

Which of the attempts at method overloading will compile without error?

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

    (A) I only

    (B) II only

    (C) III only

    (D) II and III only

    (E) I, II, and III

45. Consider the following `Point2D` class.

```
public class Point2D
{
    private double xCoord;
    private double yCoord;

    public Point2D(double x, double y)
    {
        xCoord = x;
        yCoord = y;
    }
}
```

Which of the following code segments, appearing in a class other than `Point2D,` will correctly create an instance of a `Point2D` object?

  (A) `Point2D p = (3.0, 4.0);`

  (B) `Point2D p = Point2D(3.0, 4.0);`

  (C) `new p = Point2D(3.0, 4.0);`

  (D) `new Point2D = p(3.0, 4.0);`

  (E) `Point2D p = new Point2D(3.0, 4.0);`

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

46. Consider the following class definition.

```
public class Points
{
    private double num1;
    private double num2;
    public Points(int n1, int n2)              // Line 6
    {
        num1 = n1;                             // Line 8
        num2 = n2;                             // Line 9
    }
    public void incrementPoints(int value)    // Line 12
    {
        n1 += value;                           // Line 14
        n2 += value;                           // Line 15
    }
}
```

The class does not compile. Which of the following identifies the error in the class definition?

(A) In line 6, the `Points` constructor must have a `void` return type.

(B) In lines 8 and 9, `int` values cannot be assigned to `double` variables.

(C) In line 12, the `incrementPoints` method must have a non-`void` return type.

(D) In lines 14 and 15, the variables `n1` and `n2` are not defined.

(E) In lines 14 and 15, the variable `value` is not defined.

47. Consider the following methods, which appear in the same class.

```
public void printSum(int x, double y)
{
    System.out.println(x + y);
}

public void printProduct(double x, int y)
{
    System.out.println(x * y);
}
```

Consider the following code segment, which appears in a method in the same class as `printSum` and `printProduct`.

```
int num1 = 5;
double num2 = 10.0;
printSum(num1, num2);
printProduct(num1, num2);
```

What, if anything, is printed as a result of executing the code segment?

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

(A) 15
    50

(B) 15
    50.0

(C) 15.0
    50

(D) 15.0
    50.0

(E) Nothing is printed because the code does not compile.

48. Consider the `processWords` method. Assume that each of its two parameters is a `String` of length two or more.

```
public void processWords(String word1, String word2)
{
    String str1 = word1.substring(0, 2);
    String str2 = word2.substring(word2.length() - 1);
    String result = str2 + str1;
    System.out.println(result.indexOf(str2));
}
```

Which of the following best describes the value printed when `processWords` is called?

(A) The value `0` is always printed.

(B) The value `1` is always printed.

(C) The value `result.length() - 1` is printed.

(D) A substring containing the last character of `word2` is printed.

(E) A substring containing the last two characters of `word2` is printed.

49. Consider the following method.

```
public double puzzle(int x)
{
    Double y = x / 2.0;
    y /= 2;

    return y.doubleValue();
}
```

Assume that the method call `puzzle(3)` appears in a method in the same class as `puzzle`. What value is returned as a result of the method call?

(A) 0.0

(B) 0.5

(C) 0.75

(D) 1.0

(E) 1.5

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

50. Which of the following statements assigns a random integer between 25 and 60, inclusive, to `rn` ?

    (A) `int rn = (int) (Math.random() * 25) + 36;`

    (B) `int rn = (int) (Math.random() * 25) + 60;`

    (C) `int rn = (int) (Math.random() * 26) + 60;`

    (D) `int rn = (int) (Math.random() * 36) + 25;`

    (E) `int rn = (int) (Math.random() * 60) + 25;`

51. Which of the following statements assigns a random integer between `1` and `10`, inclusive, to `rn` ?

    (A) `int rn = (int) (Math.random()) * 10;`

    (B) `int rn = (int) (Math.random()) * 10 + 1;`

    (C) `int rn = (int) (Math.random() * 10);`

    (D) `int rn = (int) (Math.random() * 10) + 1;`

    (E) `int rn = (int) (Math.random() + 1) * 10;`

52. Consider the following code segment, which is intended to assign to `num` a random integer value between `min` and `max`, inclusive. Assume that `min` and `max` are integer variables and that the value of `max` is greater than the value of `min`.

    ```
    double rn = Math.random();
    int num = /* missing code */;
    ```

    Which of the following could be used to replace /* *missing code* */ so that the code segment works as intended?

    (A) `(int) (rn * max) + min`

    (B) `(int) (rn * max) + min - 1`

    (C) `(int) (rn * (max - min)) + min`

    (D) `(int) (rn * (max - min)) + 1`

    (E) `(int) (rn * (max - min + 1)) + min`

53. Consider the following code segment. Assume that `a` is greater than zero.

    ```
    int a = /* value not shown */;
    int b = a + (int) (Math.random() * a);
    ```

    Which of the following best describes the value assigned to `b` when the code segment is executed?

    (A) `a`

    (B) `2 * a`

    (C) A random integer between `0` and `a - 1`, inclusive

    (D) A random integer between `a` and `2 * a`, inclusive

    (E) A random integer between `a` and `2 * a - 1`, inclusive

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

54. Assume that the following variable declarations have been made.

double d = Math.random();

double r;

Which of the following assigns a value to r from the uniform distribution over the range $0.5 \leq r < 5.5$?

(A) r = d + 0.5;

(B) r = d + 0.5 * 5.0;

(C) r = d * 5.0;

(D) r = d * 5.0 + 0.5;

(E) r = d * 5.5;

55. Consider the following method.

```
public static String rearrange(String str)
{
    String temp = "";
    for (int i = str.length() - 1; i > 0; i--)
    {
        temp += str.substring(i - 1, i);
    }
    return temp;
}
```

What, if anything, is returned by the method call `rearrange("apple")` ?

(A) `"appl"`

(B) `"apple"`

(C) `"elppa"`

(D) `"lppa"`

(E)   Nothing is returned due to a run-time error.

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

56. Consider the following method.

```
public static String abMethod(String a, String b)
{
    int x = a.indexOf(b);
    while (x >= 0)
    {
        a = a.substring(0, x) + a.substring(x + b.length());
        x = a.indexOf(b);
    }
    return a;
}
```

What, if anything, is returned by the method call `abMethod("sing the song", "ng")` ?

(A) `"si"`

(B) `"si the so"`

(C) `"si the song"`

(D) `"sig the sog"`

(E) Nothing is returned because a `StringIndexOutOfBoundsException` is thrown.

Directions: Select the choice that best fits each statement. The following question(s) refer to the following information.

Consider the following partial class declaration.

```
public class SomeClass
{
    private int myA;
    private int myB;
    private int myC;

    // Constructor(s) not shown

    public int getA()
    {   return myA;   }

    public void setB(int value)
    {   myB = value;   }
}
```

57. The following declaration appears in another class.
SomeClass obj = new SomeClass ( );
Which of the following code segments will compile without error?

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

(A)   int x = obj.getA ( );

(B)   int x;
       obj.getA (x);

(C)   int x = obj.myA;

(D)   int x = SomeClass.getA ( );

(E)   int x = getA(obj);

---

58. A student has created a `Song` class. The class contains the following variables.

A `String` variable called `artist` to represent the artist name
A `String` variable called `title` to represent the song title
A `String` variable called `album` to represent the album title

The object `happyBirthday` will be declared as type `Song`.

Which of the following statements is true?

(A)   `artist, title,` and `album` are instances of the `Song` class.

(B)   `happyBirthday` is an instance of three `String` objects.

(C)   `happyBirthday` is an instance of the `Song` class.

(D)   `Song` is an instance of the `happyBirthday` object.

(E)   `Song` is an instance of three `String` objects.

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

59. Consider the following class definition.

```
public class Student
{
    private int studentID;
    private int gradeLevel;
    private boolean honorRoll;

    public Student(int s, int g)
    {
        studentID = s;
        gradeLevel = g;
        honorRoll = false;
    }

    public Student(int s)
    {
        studentID = s;
        gradeLevel = 9;
        honorRoll = false;
    }
}
```

Which of the following code segments would successfully create a new `Student` object?

```
  I. Student one = new Student(328564, 11);
 II. Student two = new Student(238783);
III. int id = 392349;
     int grade = 11;
     Student three = new Student(id, grade);
```

(A) I only

(B) II only

(C) III only

(D) I and II only

(E) I, II, and III

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

60. Consider the following class definition.

```
public class Thing
{
    public void talk()
    {
        System.out.print("Hello ");
    }

    public void name()
    {
        System.out.print("my friend");
    }

    public void greet()
    {
        talk();
        name();
    }
    /* Constructors not shown */
}
```

Which of the following code segments, if located in a method in a class other than Thing, will cause the message "Hello my friend" to be printed?

(A)
```
Thing a = new Thing();
Thing.talk();
Thing.name();
```

(B)
```
Thing a = new Thing();
Thing.greet();
```

(C)
```
Thing a = new Thing();
a.talk();
```

(D)
```
Thing a = new Thing();
a.greet();
```

(E)
```
Thing a = new Thing();
a.name();
a.talk();
```

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

61. Consider the following method.

```
public int timesTwo (int n)
{
     return n * 2;
}
```

The following code segment appears in a method in the same class as the timesTwo method.

```
Integer val = 10;
int result1 = timesTwo(val);
Integer result2 = result1;
System.out.print(result2);
```

What, if anything, is printed as a result of executing the code segment?

(A) 10

(B) 20

(C) Nothing; the code segment will not compile because timesTwo cannot accept an Integer parameter.

(D) Nothing; the code segment will not compile because the value returned by timesTwo cannot be assigned to result1.

(E) Nothing; the code segment will not compile because the int variable result1 cannot be assigned to the Integer variable result2.

62. Consider the following class.

```
public class WindTurbine
{
     private double efficiencyRating;
     public WindTurbine()
     {
         efficiencyRating = 0.0;
     }
     public WindTurbine(double e)
     {
         efficiencyRating = e;
     }
}
```

Which of the following code segments, when placed in a method in a class other than WindTurbine, will construct a WindTurbine object wt with an efficiencyRating of 0.25 ?

Objects 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9

(A) `WindTurbine wt = new WindTurbine(0.25);`

(B) `WindTurbine wt = 0.25;`

(C)
```
WindTurbine wt = new WindTurbine();
wt = 0.25;
```

(D)
```
WindTurbine wt = new WindTurbine();
wt.efficiencyRating = 0.25;
```

(E) `new WindTurbine wt = 0.25;`