

U5-10\_6

Name \_\_\_\_\_

1. Consider the following class definitions.

```
public class Bike
{
    private int numWheels = 2;
    // No constructor defined
}

public class EBike extends Bike
{
    private int numBatteries;
    public EBike(int batteries)
    {
        numBatteries = batteries;
    }
}
```

The following code segment appears in a method in a class other than Bike or EBike.

```
EBike eB = new EBike(4);
```

Which of the following best describes the effect of executing the code segment?

- ☐ A An implicit call to the zero-parameter Bike constructor initializes the instance variable numWheels. The instance variable numBatteries is initialized using the value of the parameter batteries.
- ☐ B An implicit call to the one-parameter Bike constructor with the parameter passed to the EBike constructor initializes the instance variable numWheels. The instance variable numBatteries is initialized using the value of the parameter batteries.
- ☐ C Because super is not explicitly called from the EBike constructor, the instance variable numWheels is not initialized. The instance variable numBatteries is initialized using the value of the parameter batteries.
- ☐ D The code segment will not execute because the Bike class is a superclass and must have a constructor.
- ☐ E The code segment will not execute because the constructor of the EBike class is missing a second parameter to use to initialize the numWheels instance variable.



**U5-10\_6**

---

2. Consider the following class definition.

```
public class Password
{
    private String password;
    public Password (String pwd)
    {
        password = pwd;
    }
    public void reset(String new_pwd)
    {
        password = new_pwd;
    }
}
```

Consider the following code segment, which appears in a method in a class other than Password. The code segment does not compile.

```
Password p = new Password("password");
System.out.println("The new password is " + p.reset("password"));
```

Which of the following best identifies the reason the code segment does not compile?

- (A) The code segment attempts to access the private variable password from outside the Password class.
  - (B) The new password cannot be the same as the old password.
  - (C) The Password class constructor is invoked incorrectly.
  - (D) The reset method cannot be called from outside the Password class.
  - (E) The reset method does not return a value that can be printed.
- 



**U5-10\_6**

---

3. Consider the following class declaration.

```
public class Circle
{
    private double radius;
    public double computeArea()
    {
        private double pi = 3.14159;
        public double area = pi * radius * radius;
        return area;
    }
    // Constructor not shown.
}
```

Which of the following best explains why the computeArea method will cause a compilation error?

- (A) Local variables declared inside a method cannot be declared as public or private.
  - (B) Local variables declared inside a method must all be private.
  - (C) Local variables declared inside a method must all be public.
  - (D) Local variables used inside a method must be declared at the end of the method.
  - (E) Local variables used inside a method must be declared before the method header.
- 



**U5-10\_6**

---

4. Consider the following method, `count`, which is intended to traverse all the elements in the two-dimensional (2D) String array `things` and return the total number of elements that contain at least one "a".

```
public static int count(String[][] things)
{
    int count = 0;
    for (int r = 0; r < things.length; r++)
    {
        for (int c = 0; c < things[r].length - 1; c++)
        {
            if (things[r][c].indexOf("a") >= 0)
            {
                count++;
            }
        }
    }
    return count;
}
```

For example, if `things` contains `{{"salad", "soup"}, {"water", "coffee"}}`, then `count(things)` should return 2.

The method does not always work as intended. For which of the following two-dimensional array input values does `count` NOT work as intended?

- ☐ A `{{"lemon"}, {"lime"}}`
- ☐ B `{{"tall", "short"}, {"up", "down"}}`
- ☐ C `{{"rabbit", "bird"}, {"cat", "dog"}, {"gecko", "turtle"}}`
- ☐ D `{{"scarf", "gloves", "hat"}, {"shoes", "shirt", "pants"}}`
- ☐ E `{{"math", "english", "physics"}, {"golf", "baseball", "soccer"}}`
- 



**U5-10\_6**

---

5. The method `countTarget` below is intended to return the number of times the value `target` appears in the array `arr`. The method may not work as intended.

```
public int countTarget(int[] arr, int target)
{
    int count = 0;
    for (int j = 0; j <= arr.length; j++) // line 4
    {
        if (arr[j] == target)
        {
            count++;
        }
    }
    return count;
}
```

Which of the following changes, if any, can be made to line 4 so that the method will work as intended?

- (A) Changing `int j = 0;` to `int j = 1;`
  - (B) Changing `j <= arr.length;` to `j < arr.length;`
  - (C) Changing `j <= arr.length;` to `j < arr.length - 1;`
  - (D) Changing `j <= arr.length;` to `j < arr.length + 1;`
  - (E) No change is necessary; the method works correctly as is.
- 



**U5-10\_6**

---

6. Consider the following method, which is intended to return true if 0 is found in its two-dimensional array parameter `arr` and false otherwise. The method does not work as intended.

```
public boolean findZero(int[][] arr)
{
    for (int row = 0; row <= arr.length; row++)
    {
        for (int col = 0; col < arr[0].length; col++)
        {
            if (arr[row][col] == 0)
            {
                return true;
            }
        }
    }
    return false;
}
```

Which of the following values of `arr` could be used to show that the method does not work as intended?

- (A) `{{30, 20}, {10, 0}}`
- (B) `{{4, 3}, {2, 1}, {0, -1}}`
- (C) `{{0, 1, 2}, {3, 4, 5}, {6, 7, 8}}`
- (D) `{{5, 10, 15, 20}, {25, 30, 35, 40}}`
- (E) `{{10, 20, 0, 30, 40}, {60, 0, 70, 80, 90}}`
- 



**U5-10\_6**

---

7. Consider the following class definition.

```
public class Info
{
    private String name;
    private int number;
    public Info(String n, int num)
    {
        name = n;
        number = num;
    }
    public void changeName(String newName)
    {
        name = newName;
    }
    public int addNum(int n)
    {
        num += n;
        return num;
    }
}
```

Which of the following best explains why the class will not compile?

- ☐ (A) The class is missing an accessor method.
  - ☐ (B) The instance variables name and number should be designated public instead of private.
  - ☐ (C) The return type for the Info constructor is missing.
  - ☐ (D) The variable name is not defined in the changeName method.
  - ☐ (E) The variable num is not defined in the addNum method.
- 



**U5-10\_6**

---

8. Consider the following class definition.

```
public class ItemInventory
{
    private int numItems;
    public ItemInventory(int num)
    {
        numItems = num;
    }
    public updateItems(int newNum)
    {
        numItems = newNum;
    }
}
```

Which of the following best identifies the reason the class does not compile?

- (A) The constructor header is missing a return type.
  - (B) The updateItems method is missing a return type.
  - (C) The constructor should not have a parameter.
  - (D) The updateItems method should not have a parameter.
  - (E) The instance variable numItems should be public instead of private.
- 





**U5-10\_6**

---

9. Consider the following code segment, which is intended to print the sum of all elements of an array.

```
int[] arr = {10, 5, 1, 20, 6, 25};  
int sum = 0;  
for (int k = 0; k <= arr.length; k++)  
{  
    sum += arr[k];  
}  
System.out.println("The sum is " + sum);
```

A runtime error occurs when the code segment is executed. Which of the following changes should be made so that the code segment works as intended?

- (A) The for loop header should be replaced with `for (int k = 0; k < arr.length; k++)`.
  - (B) The for loop header should be replaced with `for (int k = 0; k <= arr.length; k--)`.
  - (C) The for loop header should be replaced with `for (int k = 1; k <= arr.length - 1; k++)`.
  - (D) The statement in the body of the for loop should be replaced with `sum += arr[0]`.
  - (E) The statement in the body of the for loop should be replaced with `sum += arr[k - 1]`.
- 



**U5-10\_6**

---

10. Consider the following method, `remDups`, which is intended to remove duplicate consecutive elements from `nums`, an `ArrayList` of integers. For example, if `nums` contains `{1, 2, 2, 3, 4, 3, 5, 5, 6}`, then after executing `remDups(nums)`, `nums` should contain `{1, 2, 3, 4, 3, 5, 6}`.

```
public static void remDups(ArrayList<Integer> nums)
{
    for (int j = 0; j < nums.size() - 1; j++)
    {
        if (nums.get(j).equals(nums.get(j + 1)))
        {
            nums.remove(j);
            j++;
        }
    }
}
```

The code does not always work as intended. Which of the following lists can be passed to `remDups` to show that the method does NOT work as intended?

- (A) `{1, 1, 2, 3, 3, 4, 5}`
- (B) `{1, 2, 2, 3, 3, 4, 5}`
- (C) `{1, 2, 2, 3, 4, 4, 5}`
- (D) `{1, 2, 2, 3, 4, 5, 5}`
- (E) `{1, 2, 3, 3, 4, 5, 5}`
- 



**U5-10\_6**

---

11. The `removeElement` method is intended to remove all instances of `target` from the `ArrayList` object `data` passed as a parameter. The method does not work as intended for all inputs.

```
public void removeElement(ArrayList<Integer> data, int target)
{
    for (int j = 0; j < data.size(); j++)
    {
        if (data.get(j).equals(target))
        {
            data.remove(j);
        }
    }
}
```

Assume that the `ArrayList` object `scores` and the `int` variable `low_score` have been properly declared and initialized. In which of the following cases will the method call `removeElement(scores, low_score)` fail to produce the intended result?

- (A) When `scores` is `[0, 2, 0, 2, 0, 6]` and `low_score` is 0
  - (B) When `scores` is `[2, 4, 0, 5, 7, 0]` and `low_score` is 0
  - (C) When `scores` is `[3, 4, 5, 7, 7, 2]` and `low_score` is 1
  - (D) When `scores` is `[8, 8, 4, 3, 3, 6]` and `low_score` is 3
  - (E) When `scores` is `[9, 9, 5, 9, 7, 7]` and `low_score` is 5
- 



**U5-10\_6**

---

12. Consider the following code segment, where `num` is a properly declared and initialized integer variable. The code segment is intended to traverse a two-dimensional (2D) array `arr` looking for a value equal to `num` and then print the value. The code segment does not work as intended.

```
int[][] arr = {{7, 3, 6, 4},
{9, 2, 0, 5},
{1, 4, 3, 8}};
for (int j = 0; j < arr.length - 1; j++)
{
    for (int k = 0; k < arr[0].length; k++)
    {
        if (arr[j][k] == num)
        {
            System.out.println(arr[j][k]);
        }
    }
}
```

For which of the following values of `num` does the code segment not work as intended?

- ☐ (A) `num = 5`
  - ☐ (B) `num = 6`
  - ☐ (C) `num = 7`
  - ☐ (D) `num = 8`
  - ☐ (E) `num = 9`
- 



**U5-10\_6**

---

13. Consider the following Bugs class, which is intended to simulate variations in a population of bugs. The population is stored in the method's int attribute. The getPopulation method is intended to allow methods in other classes to access a Bugs object's population value; however, it does not work as intended.

```
public class Bugs
{
    private int population;

    public Bugs(int p)
    {
        population = p;
    }

    public int getPopulation()
    {
        return p;
    }
}
```

Which of the following best explains why the getPopulation method does NOT work as intended?

- (A) The getPopulation method should be declared as private.
  - (B) The return type of the getPopulation method should be void.
  - (C) The getPopulation method should have at least one parameter.
  - (D) The variable population is not declared inside the getPopulation method.
  - (E) The instance variable population should be returned instead of p, which is local to the constructor.
- 



**U5-10\_6**

---

14. In the following code segment, assume that the ArrayList numList has been properly declared and initialized to contain the Integer values [1, 2, 2, 3]. The code segment is intended to insert the Integer value val in numList so that numList will remain in ascending order. The code segment does not work as intended in all cases.

```
int index = 0;
while (val > numList.get(index))
{
    index++;
}
numList.add(index, val);
```

For which of the following values of val will the code segment not work as intended?

- (A) 0
  - (B) 1
  - (C) 2
  - (D) 3
  - (E) 4
- 



**U5-10\_6**

---

15. Consider the following class definitions.

```
public class Computer
{
    private String memory;
    public Computer()
    {
        memory = "RAM";
    }
    public Computer(String m)
    {
        memory = m;
    }
    public String getMemory()
    {
        return memory;
    }
}

public class Smartphone extends Computer
{
    private double screenWidth, screenHeight;
    public SmartPhone(double w, double h)
    {
        super("flash");
        screenWidth = w;
        screenHeight = h;
    }
    public double getScreenWidth()
    {
        return screenWidth;
    }
    public double getScreenHeight()
    {
        return screenHeight;
    }
}
```

The following code segment appears in a class other than Computer or Smartphone.

```
Computer myPhone = new SmartPhone(2.55, 4.53);
System.out.println("Device has memory: " + myPhone.getMemory() +
    ", screen area: " + myPhone.getScreenWidth() *
    myPhone.getScreenHeight() + " square inches.");
```

The code segment is intended to produce the following output.

Device has memory: flash, screen area: 11.5515 square inches.

Which of the following best explains why the code segment does not work as intended?



**U5-10\_6**

---

- (A) An error occurs during compilation because a Smartphone object cannot be assigned to the Computer reference variable myPhone.
- (B) An error occurs during compilation because the Smartphone class has no getMemory method.
- (C) An error occurs during compilation because the getScreenWidth and getScreenHeight methods are not defined for the Computer object myPhone.
- (D) An error occurs at runtime because the Smartphone class has no getMemory method.
- (E) An error occurs at runtime because the getScreenWidth and getScreenHeight methods are not defined for the Computer object myPhone.
- 

**16.** Consider the following class definitions.

```
public class C1
{
    public C1()
    { /* implementation not shown */ }
    public void m1()
    { System.out.print("A"); }
    public void m2()
    { System.out.print("B"); }
}
public class C2 extends C1
{
    public C2()
    { /* implementation not shown */ }
    public void m2()
    { System.out.print("C"); }
}
```

The following code segment appears in a class other than C1 or C2.

```
C1 obj1 = new C2();
obj1.m1();
obj1.m2();
```

The code segment is intended to produce the output AB. Which of the following best explains why the code segment does not produce the intended output?





**U5-10\_6**

---

- (A) A compile-time error occurs because obj1 is declared as type C1 but instantiated as type C2.
  - (B) A runtime error occurs because method m1 does not appear in C2.
  - (C) Method m1 is not executed because it does not appear in C2.
  - (D) Method m2 is executed from the subclass instead of the superclass because obj1 is instantiated as a C2 object.
  - (E) Method m2 is executed twice (once in the subclass and once in the superclass) because it appears in both classes.
- 



**U5-10\_6**

---

17. Consider the following two class definitions.

```
public class Bike
{
    private int numOfWheels = 2;
    public int getNumOfWheels()
    {
        return numOfWheels;
    }
}

public class EBike extends Bike
{
    private int numOfWatts;
    public EBike(int watts)
    {
        numOfWatts = watts;
    }
    public int getNumOfWatts()
    {
        return numOfWatts;
    }
}
```

The following code segment occurs in a class other than Bike or EBike.

```
Bike b = new EBike(250);
System.out.println(b.getNumOfWatts());
System.out.println(b.getNumOfWheels());
```

Which of the following best explains why the code segment does not compile?

- (A) The Bike superclass does not have a constructor.
  - (B) There are too many arguments to the EBike constructor call in the code segment.
  - (C) The first line of the subclass constructor is not a call to the superclass constructor.
  - (D) The getNumOfWatts method is not found in the Bike class.
  - (E) The getNumOfWheels method is not found in the EBike class.
- 



**U5-10\_6**

---

18. Consider the following class definition.

```
public class FishTank
{
    private double numGallons;
    private boolean saltWater;
    public FishTank(double gals, boolean sw)
    {
        numGallons = gals;
        saltWater = sw;
    }
    public double getNumGallons()
    {
        return numGallons;
    }
    public boolean isSaltWater()
    {
        if (saltWater)
        {
            return "Salt Water";
        }
        else
        {
            return "Fresh Water";
        }
    }
}
```

Which of the following best explains the reason why the class will not compile?



**U5-10\_6**

---

- (A) The variable numGallons is not declared in the getNumGallons method.
  - (B) The variable saltWater is not declared in the isSaltWater method.
  - (C) The isSaltWater method does not return the value of an instance variable.
  - (D) The value returned by the getNumGallons method is not compatible with the return type of the method.
  - (E) The value returned by the isSaltWater method is not compatible with the return type of the method.
- 

**19.** Consider the code segment below, where arr is a one-dimensional array of integers.

```
int sum = 0;
for (int n : arr)
{
    sum = sum + 2 * n;
}
System.out.print(sum);
```

Which of the following code segments will produce the same output as the code segment above?



**U5-10\_6**

---

```
int sum = 0;
for (int k = 0; k < arr.length; k++)
```

**(A)**

```
{
    sum = sum + 2 * k;
}
System.out.print(sum);
```

```
int sum = 0;
for (int k = 0; k <= arr.length; k++)
```

**(B)**

```
{
    sum = sum + 2 * k;
}
System.out.print(sum);
```

```
int sum = 0;
for (int k = 1; k <= arr.length; k++)
```

**(C)**

```
{
    sum = sum + 2 * k;
}
System.out.print(sum);
```

```
int sum = 0;
for (int k = 0; k < arr.length; k++)
```

**(D)**

```
{
    sum = sum + 2 * arr[k];
}
System.out.print(sum);
```

```
int sum = arr[0];
for (int k = 1; k <= arr.length; k++)
```

**(E)**

```
{
    sum = sum + 2 * arr[k];
}
System.out.print(sum);
```

---

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

Consider the following instance variable and methods. You may assume that data has been initialized with `length > 0`. The methods are intended to return the index of an array element equal to `target`, or `-1` if no such element exists.



**U5-10\_6**

---

```
private int[] data;

public int seqSearchRec(int target)
{
    return seqSearchRecHelper(target, data.length - 1);
}

private int seqSearchRecHelper(int target, int last)
{
    // Line 1

    if (data[last] == target)
        return last;
    else
        return seqSearchRecHelper(target, last - 1);
}
```

---

**20.** For which of the following test cases will the call `seqSearchRec(5)` always result in an error?

1.  
data contains only one element.
2.  
data does not contain the value 5.
3.  
data contains the value 5 multiple times.

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