Started on	Monday, 26 March 2018, 9:44 PM
State	Finished
Completed on	Monday, 26 March 2018, 10:07 PM
Time taken	23 mins 50 secs
Marks	49.00/50.00
Grade	**98.00** out of 100.00

Question 1

Correct

Mark 1.00 out of 1.00

Fill in the blank to make the code compile: package animal;

```
package animal;

public class Cat {

    public String name;

    public static void main(String[] meow) {

        Cat cat= new Cat();

        _____= "Sadie";

    }

}
```

Select one:

○ a. cat[name]

○ b. cat.name

○ c. cat-name

◉ d. cat.setName ✓ Java uses dot notation to reference instance variables in a class, making Option A correct.

The correct answer is: cat.setName

Question 2

Correct

Mark 1.00 out of 1.00

Fill in the blanks to indicate whether a primitive or wrapper class can be assigned without the compiler using the autoboxing feature.
_____first= Integer.parseint("5");
_____second= Integer.valueOf("5");

Select one:

- ○ a. int, int

- ○ b. Integer,int

- ● c. int, Integer ✔ The parseInt() methods return a primitive. The valueOf() methods return a wrapper class object. In real code, autoboxing would let you assign the return value to either a primitive or wrapper class. In terms of what gets returned directly,

- ○ d. Integer, Integer

The correct answer is: int, Integer

Question 3

Correct

Mark 1.00 out of 1.00

Given the following code, fill in the blank to have the code print bounce.

```
public class TennisBall {

    public TennisBall() {

    System.out.println("bounce");

    }

    public static void main(String[] slam) {

        _____

    }

}
```

Select one:

- ○ a. TennisBall;

- ○ b. TennisBall();

- ● c. new TennisBall(); ✔ In order to call a constructor, you must use the new keyword. It cannot be called as if it was a normal method.

- ○ d. new TennisBall;

The correct answer is: new TennisBall();

Question 4

Correct

Mark 1.00 out of 1.00

How do you force garbage collection to occur at a certain point?

Select one:

- a. Call System. requireGc()
- b. Call System.gc()
- c. None of the above ✓ While you can suggest to the JVM that it might want to run a garbage collection cycle, the JVM is free to ignore your suggestion.
- d. Call System.forceGc()

The correct answer is: None of the above

How many instance initializers are in this code?

```java
public class Bowling {

    {
        System.out.println();
    }

    public Bowling() {

        System.out.println();

    }

    static {
        System.out.println();
    }

    {
        System.out.println();
    }

}
```

Select one:

- ⦿ a. Two ✓   Lines 2 and 7 illustrate instance initializers. Line 6 is a static initializer. Lines 3–5 are a constructor
- ○ b. Three
- ○ c. One
- ○ d. None

The correct answer is: Two

How many objects are eligible for garbage collection right before the end of the main method?

```
public class Person {
2:     public Person youngestChild;

3:

4:     public static void main(String ... args) {

5:          Person elena = new Person();

6:     Person diana = new Person();

7:     elena.youngestChild = diana;

8:     diana = null;

9:     Person zoe = new Person();

10:    elena.youngestChild = zoe;

11:    zoe = null;

12:    }

13:}
```

Select one:

○ a. Three

○ b. None

◉ c. One ✔ On line 9, all three objects have references. The elena and zoe objects have a direct reference. The diana object is referenced through the elena object. On line 10, the reference to the diana object is replaced by a reference to the zoe object. Therefore, the diana object is eligible to be garbage collected.

○ d. Two

The correct answer is: One

How many of the following lines compile?

```
int i = null;

Integer in = null;

Strings = null;
```

Select one:

- a. None

- b. Two ✓ Objects are allowed to have a null reference while primitives cannot. int is a primitive, so assigning null to it does not compile. Integer and String are both objects and can therefore be assigned a null reference.

- c. Three

- d. One

The correct answer is: Two

How many of the following methods compile?

```
public String convert(int value){
        return value.toString();
        }
public String convert(Integer value){
        return value.toString();
        }
public String convert(Object value){
        return value.toString();
        }
```

Select one:

- a. Three

- b. One

- c. Two ✓ Objects have instance methods while primitives do not. Since int is a primitive, you cannot call instance methods on it. Integer and String are both objects and have instance methods.

- d. None

The correct answer is: Two

Question 9

Correct

Mark 1.00 out of 1.00

How many of the string objects are eligible for garbage collection right before the end of the main method?

```
public static void main (String[] fruits) {

String fruitl = new String("apple");

String fruit2 = new String("orange");

String fruit3 = new String("pear");

fruit3 = fruit1;

fruit2 = fruit3;

fruit1 = fruit2;

}
```

Select one:

○ a. None

◉ b. Two ✓   All three references point to the String apple. This makes the other two String objects eligible for garbage collection and Option C correct.

○ c. One

○ d. Three

The correct answer is: Two

Question 10

Correct

Mark 1.00 out of 1.00

```
Of the types double, int, and short, how many could fill in the
blank to have this code output 0?

public static void main (String[] args) {

      defaultValue; System.out.println(defaultValue);

}
```

Select one:

◉ a. None ✓   Since defaultValue is a local variable, it is not automatically initialized. That means the code will not compile with any type. Therefore, Option A is correct. If this was an instance variable, Option C would be correct as int and short would be initialized to 0 while double would be initialized to 0.0.

○ b. One

○ c. Two

○ d. Three

The correct answer is: None

Question 11

Correct

Mark 1.00 out of 1.00

Of the types double, int, long, and short, how many could fill in the blank to have this code output 0?

Static _____defaultValue;
public static void main(String[] args) {
System.out.println(defaultValue);
}

Select one:

   ○ a. Four

   ○ b. One

   ○ c. Two

   ⦿ d. Three ✓ Since defaultValue is an instance variable, it is automatically initialized to the corresponding value for that type. For double, that value is 0.0. By contrast, it is 0 for int, long, and short.

The correct answer is: Three

Question 12

Correct

Mark 1.00 out of 1.00

Suppose foo is a reference to an instance of a class. Which of the following is not true about foo. bar?

Select one:

   ○ a. bar is an instance variable.

   ○ b. It can be used to write to bar.

   ⦿ c. bar is a local variable. ✓ Dot notation is used for both reading and writing instance variables, assuming they are in scope. It cannot be used for referencing local variables,

   ○ d. It can be used to read from bar.

The correct answer is: bar is a local variable.

Suppose you have the following code. Which of the images best represents the state of the references right before the end of the main method, assuming garbage collection hasn't run?

```
1:    public class Link {
2:        private String name;
3:        private Link next;
4:        public Link(String name, Link next) {
5:        this.name= name;
6:        this.next= next;
7:    }
8:    public void setNext(Link next) {
9:        this.next= next;
10:    }
11:    public Link getNext() {
12:        return next;
13:     }
14:    public static void main(String ... args) {
15:        Link linkl = new Link("x", null);
16:        Link link2 = new Link("y", linkl);
17:        Link link3 = new Link( "z", link2);
18:        link2.setNext(link3);
19:        link3.setNext(link2);
20:        linkl = null;
21:        link3 = null;
22:    }
23: }
```

Select one:

○ a. Option D

◉ b. Option C ✓ Lines 15–17 create the three objects. Lines 18–19 change the references so link2 and link3 point to each other. The lines 20–21wipe out two of the original references. This means the object with name as x is inaccessible.

○ c. Option A

○ d. Option B

The correct answer is: Option C

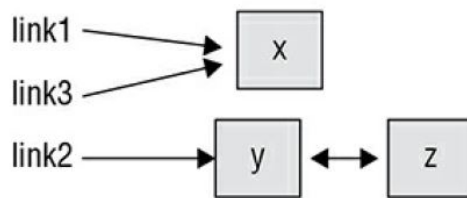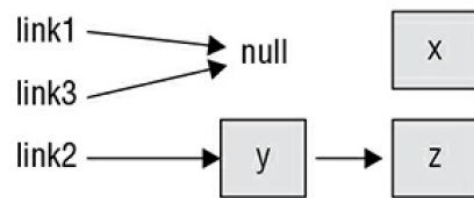What does the following output?

```
1: public class InitOrder {
2:    public String first = "instance";
3:    public InitOrder() {
4:       first = "constructor";
5:    }
6:    { first = "block"; }
7:    public void print() {
8:        System.out.println(first);
9:    }
10:  public static void main(String... args) {
11:      new InitOrder().print();
12:  }
13: }
```
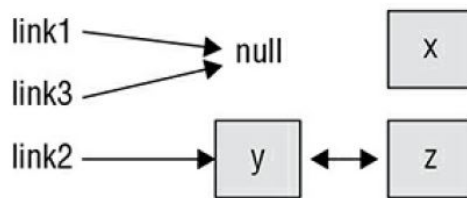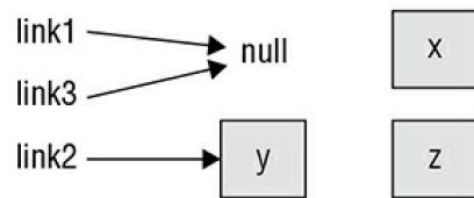
Option A

link1 ─────▶ [X]
link3 ─────▶

link2 ─────▶ [y] ◀──▶ [z]

Option B

link1 ─────▶ null   [X]
link3 ─────▶

link2 ─────▶ [y] ──▶ [z]

Option C

link1 ─────▶ null   [X]
link3 ─────▶

link2 ─────▶ [y] ◀──▶ [z]

Option D

link1 ─────▶ null   [X]
link3 ─────▶

link2 ─────▶ [y]   [z]

Select one:

○ a. instance

◉ b. constructor ✓ First line 2 runs and sets the variable using the declaration. Then the instance initializer on line 6 runs. Finally, the constructor runs. Since the constructor is the last to run of the three, that is the value that is set when we print the result

○ c. block

○ d. The code does not compile.

The correct answer is: constructor

Question **15**

Correct

Mark 1.00 out of 1.00

What is the first line in the following code to not compile?

```
public static void main(String[] args) {
    int   Integer  - 0;     //k1
    Integer int= 0;        //k2
    Integer++;             //k3
    int++;                 //k4
}
```

Select one:

○ a. kl

◉ b. k2 ✓   Integer is the name of a class in Java. While it is bad practice to use the name of a class as your local variable name, this is legal. Therefore, k1 does compile. It is not legal to use a reserved word as a variable name. All of the primitives including int are reserved words. Therefore, k2 does not compile, and this option is the answer. Line k4 doesn't compile either, but the question asks about the first line to not compile.

○ c. k3

○ d. k4

The correct answer is: k2

What is the output of the following?

Integer integer= new Integer(4);
System.out.print(integer.byteValue());
System.out.print("-");
inti= new Integer(4);
System.out.print(i.byteValue());

Select one:

○ a. 4-4

○ b. The code does not compile.

○ c. 4-0

◉ d. The code compiles but throws an exception at runtime. ✗

The correct answer is: The code does not compile.

What is the output of the following?

```
1: public class InitOrder {
2:    public String first = "instance";
3:    public InitOrder() {
4:      first = "constructor";
5:    }
6:    { first = "block"; }
7:    public void print() {
8:       System.out.println(first);
9:    }
10:  public static void main(String... args) {
11:     new InitOrder().print();
12:    }
13: }
```

Select one:

○ a. None of the above

○ b. ab

◉ c. aab ✓   The main() method calls the constructor which outputs a. Then the main method calls the run() method. The run() method calls the constructor again, which outputs a again. Then the run() method calls the Sand() method, which happens to have the same name as the constructor. This outputs b.

○ d. a

The correct answer is: aab

What is the output of the following?

```java
public  static  void main(String ... args) {
    String  chair,    table = "metal";
    chair= chair+ table;
    System.out.println(chair);
}
```

Select one:

○ a. nullmetal

⦿ b. The code does not compile. ✓   The table variable is initialized to "metal". However, chair is not initialized. In Java, initialization is per variable and not for all the variables in a single declaration. Therefore, the second line tries to reference an uninitialized local variable and does not compile,

○ c. metalmetal

○ d. metal

The correct answer is: The code does not compile.

What is the result of running this code?

```
public class Values {

integer a=  Integer.valueOf("1");

public static void main(String[] nums) {

integer a=  Integer.valueOf("2");

        integer b = Integer.valueOf("3");

        System.out.println(a + b);

 }

}
```

Select one:

- ⦿ a. The code does not compile. ✓ There is no class named integer. There is a primitive int and a class Integer. Therefore, the code does not compile, and Option C is correct. If the type was changed to Integer, Option B would be correct

- ○ b. The code compiles but throws an exception at runtime.

- ○ c. 5

- ○ d. 4

The correct answer is: The code does not compile.

What is true of the finalize() method?

Select one:
- ⦿ a. It may be called zero or one times. ✓ The finalize() method may not be called, such as if your
program crashes. However, it is guaranteed to be called no more than once.

- ○ b. It will be called exactly once.

- ○ c. It may be called zero or more times.

- ○ d. It may be called one or more times.

The correct answer is: It may be called zero or one times.

Question 21

Correct

Mark 1.00 out of 1.00

Which best describes what the new keyword does?

Select one:

○ a. Instantiates a new object ✓ The new keyword is used to call the constructor for a class and instantiate an instance of the class. A primitive cannot be created using the new keyword. Dealing with references happens after the object created by new is returned.

○ b. Switches an object reference to a new one

○ c. Creates a copy of an existing object and treats it as a new one

○ d. Creates a new primitive

The correct answer is: Instantiates a new object

Question 22

Correct

Mark 1.00 out of 1.00

Which converts a primitive to a wrapper class object without using autoboxing?

Select one:

○ a. Call the asobject() method

○ b. Call the constructor of the wrapper class ✓ Each wrapper class has a constructor that takes the primitive equivalent. The methods mentioned in Options A, C, and D do not exist

○ c. Call the convertToObj ect() method

○ d. Call the toObject() method

The correct answer is: Call the constructor of the wrapper class

---

Question 23

Correct

Mark 1.00 out of 1.00

Which is a valid constructor for this class? public class TennisBall { }

Select one:

○ a. public TennisBall static create() { return new TennisBall(); }

○ b. public TennisBall static newinstance() { return new TennisBall():}

○ c. public TennisBall() {} ✓ Options A and B are static methods rather than constructors. Option D is a method that happens to have the same name as the class. It is not a constructor because constructors don't have return types.

○ d. public void TennisBall() {}

The correct answer is: public TennisBall() {}

Question 24

Correct

Mark 1.00 out of 1.00

Which is correct about a local variable of type string?

Select one:

○ a. It defaults to null.

○ b. It will not compile without initializing on the declaration line.

◉ c. It does not have a default value. ✓ Local variables do not have a default initialization value. If they
are referenced before being set to a value, the code does not compile.

○ d. It defaults to an empty string.

The correct answer is: It does not have a default value.

Question 25

Correct

Mark 1.00 out of 1.00

Which is correct about an instance variable of type string?

Select one:

○ a. It defaults to an empty string.

◉ b. It defaults to null. ✓ Instance variables have a default value based on the
type. For any non-primitive, including String, that type is a reference to null.
Therefore Option B is correct. If the variable was a local variable, Option C
would be correct.

○ c. It does not have a default value.

○ d. It will not compile without initializing on the declaration line.

The correct answer is: It defaults to null.

Which is the first line to trigger a compiler error?
double dl = 5f; // pl
double d2 = 5.0; // p2
float fl - 5f; // p3
float f2 = 5.0; // p4

Select one:

○ a. pl

○ b. p3

● c. p4 ✔ Java uses the suffix f to indicate a number is a float. Java automatically widens a type, allowing a float to be assigned to either a float or a double. This makes both lines p1 and p3 compile. Line p2 does compile without a suffix. Line p4 does not compile without a suffix and therefore is the answer.

○ d. p2

The correct answer is: p4

Which is the most common way to fill in the blank to implement this method?

```
public class Penguin \{

    private double beaklength;

    public static void setBeakLength(Penguin p, int b) \{

    \}

\}
```

Select one:

● a. p.beaklength = b; ✔ The other Options do not compile. In Java, braces are for arrays rather than instance variables. This Option A is the correct answer. It uses dot notation to access the instance variable. It also shows that a private variable is accessible in the same class and that a narrower type is allowed to be assigned to a wider type.

○ b. p['beaklength'] = b;

○ c. p[beaklength] = b;

○ d. None of the above

The correct answer is: p.beaklength = b;

Which of the following can fill in the blanks to make this code compile?
_____ d = new (1_000_000_.00);

Select one:

- a. Double,double

- b. double,double

- ⦿ c. None of the above ✓ This question is tricky as it appears to be about primitive vs.
  wrapper classes. Looking closely, there is an underscore right before the decimal point. This is illegal as the underscore in a numeric literal can only appear between two digits.

- d. double, Double

The correct answer is: None of the above

Which of the following can fill in the blanks to make this code compile?

_____ d = new _____(1_000_000.00);

Select one:

- a. double,double

- b. double, Double

- c. Double,double

- ⦿ d. None of the above ✓ This question is tricky as it appears to be about primitive vs. wrapper classes. Looking closely, there is an underscore right before the decimal point. This is illegal as the underscore in a numeric literal can only appear between two digits.

The correct answer is: None of the above

Which of the following correctly assigns animal to both variables?
I. String cat = "animal", dog = "animal";
II. String cat - "animal"; dog - "animal";
III. String cat, dog = "animal";
IV. String cat, String dog = "animal";

Select one:

  ○ a. I, II, III, IV

  ○ b. I, III

  ○ c. I, II

  ⊙ d. I ✓ This option(I) correctly assigns the value to both variables. II does not
     compile as dog does not have a type. Notice the semicolon in that line, which
     starts a new statement. III compiles but only assigns the value to dog since a
     declaration only assigns to one variable rather than everything in the
     declaration. IV does not compile because the type should only be specified once
     per declaration.

The correct answer is: I

Which of the following declarations does not compile?

Select one:

  ○ a. int num1, num2 = 0;

  ○ b. int num1 = 0, num2 = 0;

  ○ c. int num1, num2;

  ⊙ d. double num1, int num2 = 0; ✓ Option A does not compile because Java
     does not allow declaring different types as part of the same declaration. The
     other three options show various legal combinations of combining multiple
     variables in the same declarations with optional default values.

The correct answer is: double num1, int num2 = 0;

Question 32

Correct

Mark 1.00 out of 1.00

Which of the following does not compile compile?

Select one:

- ○ a. int num = 999;
- ● b. int num = _9_99; ✓ Underscores are allowed between any two digits in a numeric literal. Underscores are not allowed at the beginning or end of the literal, making this Option the correct answer.
- ○ c. int num = 9_9_9;
- ○ d. None of the above; they all compile.

The correct answer is: int num = _9_99;

Question 33

Correct

Mark 1.00 out of 1.00

Which of the following does not compile?

Select one:

- ○ a. None of the above; they all compile.
- ● b. double num = 2._718; ✓ Underscores are allowed between any two digits in a numeric literal. Underscores are not allowed adjacent to a decimal point.
- ○ c. double num = 2.7_1_8;
- ○ d. double num = 2.718;

The correct answer is: double num = 2._718;

Question 34

Correct

Mark 1.00 out of 1.00

Which of the following is a wrapper class?

Select one:

- ○ a. int
- ○ b. Int
- ● c. Integer ✓ Option A is incorrect because int is a primitive. Option B is incorrect because it is not the name of a class in Java. While Option D is a class in Java, it is not a wrapper class because it does not map to a primitive.
- ○ d. Object

The correct answer is: Integer

4888888. Which of the following is not a possible output of this code, assuming it runs to completion?

```
package store;

  public class Toy {

        public void play() {

          System.out.print("play-");

        }

        public void finalize() {

        System.out.pr int("c lean- 11 )      ;

        }

        public static void main(String[] args) {

        Toy car= new Toy();

        car. play();

        System.gc();

        Toy doll= new Toy();

        doll.play();

        }

}
```

Select one:

⦿ a. play- ✓   Remember that garbage collection is not guaranteed to run on demand. If it doesn't run at all, Option B would be output. If it runs at the requested point, Option C would be output. If it runs right at the end of the main() method, Option D would be output. Option A is the correct answer because play is definitely called twice. Note that you are unlikely to see all these scenarios if you run this code because we have not used enough memory for garbage collection to
be worth running. However, you still need to be able to answer what could happen regardless of it being unlikely.

○ b. play-play-

○ c. play-play-clean-

○ d. play-play-clean-clean-

The correct answer is: play-

Question 36

Correct

Mark 1.00 out of 1.00

Which of the following is not a valid class declaration?

Select one:

○ a. class 5MainSt {} ✓ Class names follow the same requirements as other identifiers. Underscores and dollar signs are allowed. Numbers are allowed, but not as the first character of an identifier. Therefore, Option C is correct. Note that class names begin with an uppercase letter by convention, but this is not a requirement.

○ b. class building {}

○ c. class _Outside{}

○ d. class cost${}

The correct answer is: class 5MainSt {}

Question 37

Correct

Mark 1.00 out of 1.00

Which of the following is not a valid order for elements in a class?

Select one:

○ a. Instance variables, constructor, method names

○ b. Constructor, instance variables, method names

○ c. Method names, instance variables, constructor

○ d. None of the above: all orders are valid. ✓ The instance variables, constructor, and method names can appear in any order within a class declaration.

The correct answer is: None of the above: all orders are valid.

Question 38

Correct

Mark 1.00 out of 1.00

Which of the following is not a valid variable name?

Select one:

○ a. Blue

○ b. blue$

○ c. blue

○ d. 2blue ✓ An identifier name must begin with a letter, $, or _. Numbers are only permitted for subsequent characters. Therefore, Option B is not a valid variable name.

The correct answer is: 2blue

Which of the following is not a wrapper class?

Select one:

○ a. String ✓ String is a class, but it is not a wrapper class. In order to be a wrapper class, the class must have a one-to-one mapping with a primitive.

○ b. Double

○ c. Integer

○ d. Long

The correct answer is: String

Question **39**

Which of the following is not a wrapper class?

Select one:

○ a. String ✓ String is a class, but it is not a wrapper class. In order to be a wrapper class, the class must have a one-to-one mapping with a primitive.

Which of the following is the output of this code, assuming it runs to completion?

```
package store; public class Toy \{
    public void play() \{
        System.out.pr int("p lay- 11 ) ;
    \}
    public void finalizer() \{
        System.out.pr int("c lean- 11 ) ;
    \}
    public static void main(String[] fun) \{
        Toy car\= new Toy();
        car. play();
        System.gc();
        Toy doll\= new Toy();
        doll.play();
    \}
\}
```

Select one:
- ○ a. play-clean-play-
- ○ b. play-
- ○ c. play-play-clean-clean-
- ⦿ d. play-play- ✓ If there was a finalize() method, this would be a different story. However, the method here is finalizer. Tricky! That's just a normal method that doesn't get called automatically. Therefore clean is never output.

The correct answer is: play-play-

Question 41

Correct

Mark 1.00 out of 1.00

Which of the following is true about primitives?

Select one:

○ a. You can call methods on a primitive.

○ b. You can store a primitive directly into an ArrayList.

◉ c. You can convert a wrapper class object to a primitive by calling valueof (). ✔
　 This option is an example of autoboxing. Java will automatically convert from primitive to wrapper class types and vice versa.

○ d. You can convert a primitive to a wrapper class object simply by assigning it.

The correct answer is: You can convert a wrapper class object to a primitive by calling valueof ().

Question 42

Correct

Mark 1.00 out of 1.00

Which of the following is true about string instance variables?

Select one:

○ a. They can never be set from outside the class they are defined in.

◉ b. They can be set to null. ✔　 Assuming the variables are not primitives, they allow a null assignment. The other statements are false.

○ c. They can only be set once per run of the program.

○ d. They can only be set in the constructor.

The correct answer is: They can be set to null.

Question 43

Correct

Mark 1.00 out of 1.00

Which of the following lines contains a compiler error?
String title= "Weather"; // line x1
int hot, double cold; // line x2
System.out.println(hot +" "+ title); // line x3

Select one:

○ a. None of the above

⦿ b. x2 ✔ Java does not allow multiple Java data types to be declared in the same declaration, making Option B the correct answer. If double was removed, both hot and cold would be the same type. Then the compiler error would be on x3 because of a reference to an uninitialized variable.

○ c. x3

○ d. x1

The correct answer is: x2

Question 44

Correct

Mark 1.00 out of 1.00

Which of the following lists of primitive numeric types is presented in order from smallest to largest data type?

Select one:

○ a. int, short, byte, long

⦿ b. byte, short, int, long ✔ These four types represent nondecimal values. While you don't need to know the exact sizes, you do need to be able to order them from largest to smallest. A byte is smallest. A short comes next, followed by int and then long.

○ c. short, int, byte, long

○ d. short, byte, int, long

The correct answer is: byte, short, int, long

Question 45

Correct

Mark 1.00 out of 1.00

Which of the following lists of primitive types are presented in order from smallest to largest data type?

Select one:

○ a. char,double, float, bigint

◉ b. byte,char, float, double ✓   A byte is smaller than a char, making Option C incorrect. Bigint is not a primitive, making Option D incorrect. A double uses twice as much memory as a float variable.

○ c. char, byte, float, double

○ d. byte,char, double, float

The correct answer is: byte,char, float, double

Question 46

Correct

Mark 1.00 out of 1.00

Which of these class names best follows standard Java naming conventions?

Select one:

○ a. fooBar

◉ b. FooBar ✓   In Java, class names begin with an uppercase letter by convention. Then they use lowercase with the exception of new words. Option B follows this convention and is correct. Option A follows the convention for variable names. Option C follows the convention for constants. Option D doesn't follow any Java conventions.

○ c. FOO_BAR

○ d. F_o_o_B_a_r

The correct answer is: FooBar

Question 47

Correct

Mark 1.00 out of 1.00

Which pairs of statements can accurately fill in the blanks in this table?

Variable Type	Can be called within the class from what type of method
Instance	Blank 1:_____
Static	Blank 2: _____

Select one:

⦿ a. Blank 1: an instance method only, Blank 2: an instance or static method ✓
   An instance variable can only be referenced from instance methods in the class. A static variable can be referenced from any method.

◯ b. Blank 1: an instance or static method, Blank 2: a static method only

◯ c. Blank 1: an instance method only, Blank 2: a static method only

◯ d. Blank 1: an instance or static method, Blank 2: an instance or static method

The correct answer is: Blank 1: an instance method only, Blank 2: an instance or static method

Question 48

Correct

Mark 1.00 out of 1.00

Which statement is true about primitives?

Select one:

◯ a. Primitive types can be set to null.

◯ b. String is a primitive.

⦿ c. Primitive types begin with a lowercase letter. ✓ An example of a primitive type is int. All the primitive types are lowercase, making this Option correct. Unlike object reference variables, primitives cannot reference null. String is not a primitive as evidenced by the uppercase letter in the name and the fact that we can call methods on it. You can create your own classes, but not primitives.

◯ d. You can create your own primitive types.

The correct answer is: Primitive types begin with a lowercase letter.

Question 49

Correct

Mark 1.00 out of 1.00

Which two primitives have wrapper classes that are not merely the name of the primitive with an uppercase letter?

Select one:

○ a. byte and char

◉ b. char and int ✓ The wrapper class for int is Integer and the wrapper class for char is Character. All other primitives have the same name. For example, the wrapper class for boolean is Boolean

○ c. None of the above

○ d. byte and int

The correct answer is: char and int

Question 50

Correct

Mark 1.00 out of 1.00

Which type can fill in the blank?

Select one:

◉ a. double ✓ Options A and D are incorrect because byte and short do not store values with decimal points. Option B is tempting. However, 3.14 is automatically a double. It requires casting to float or writing 3.14f in order to be assigned to a float

○ b. byte

○ c. short

○ d. _____ pi= 3.14;

○ e. float

The correct answer is: double