Methods and Exception

Total points 11/39

The respondent's email (**aazadbablesh@gmail.com**) was recorded on submission of this form.

✓ 1. What modifiers are implicitly applied to all interface methods? (Choose all that apply *1/1

- [ ] A. protected
- [x] B. public ✓
- [ ] C. static
- [ ] D. void
- [ ] E. abstract
- [ ] F. default

✗ 2. What is the output of the following code? 0/1

```
1: class Mammal {
2: public Mammal(int age) {
 3: System.out.print("Mammal");
4: }
5: }
6: public class Platypus extends Mammal {
7: public Platypus() {
 8: System.out.print("Platypus");
9: }
10: public static void main(String[] args) {
11: new Mammal(5);
12: } 13: }
```

- ○ A. Platypus
- ◉ B. Mammal ✗
- ○ C. PlatypusMammal
- ○ D. MammalPlatypus
- ○ E. The code will not compile because of line 8.
- ○ F. The code will not compile because of line 11.

✗ 3. Which of the following statements can be inserted in the blank line so that the code will compile successfully? (Choose all that apply) 0/1

```
public interface CanHop {}
public class Frog implements CanHop { public static void main(String[] args) {
frog = new TurtleFrog(); } }
public class BrazilianHornedFrog extends Frog {} public class TurtleFrog extends Frog {}
```

- [ ] A. Frog
- [ ] B. TurtleFrog
- [x] C. BrazilianHornedFrog ✗
- [x] D. CanHop ✓
- [x] E. Object ✓
- [ ] F. Long

✗ 4. Which statement(s) are correct about the following code? (Choose all that apply) *0/1

public class Rodent { protected static Integer chew() throws Exception { System.out.println("Rodent is chewing"); return 1; } }
public class Beaver extends Rodent { public Number chew() throws RuntimeException { System.out.println("Beaver is chewing on wood"); return 2; } }

- [ ] A. It will compile without issue.
- [ ] B. It fails to compile because the type of the exception the method throws is a subclass of
- [ ] the type of exception the parent method throws.
- [ ] C. It fails to compile because the return types are not covariant.
- [ ] D. It fails to compile because the method is protected in the parent class and public in
- [ ] the subclass.
- [x] E. It fails to compile because of a static modifier mismatch between the two methods ✓

✗ 5. Which of the following may only be hidden and not overridden? (Choose all that apply) *0/1

- [x] A. private instance methods ✓
- [ ] B. protected instance methods
- [ ] C. public instance methods
- [ ] D. static methods
- [ ] E. public variables
- [x] F. private variables ✓

✗ 6. Choose the correct statement about the following code: 1: interface HasExoskeleton { 0/1
2: abstract int getNumberOfSections();
3: }
4: abstract class Insect implements HasExoskeleton {
5: abstract int getNumberOfLegs();
6: }
7: public class Beetle extends Insect {
8: int getNumberOfLegs() { return 6; }
9: }

- [ ] A. It compiles and runs without issue.
- [x] B. The code will not compile because of line 2. ✗
- [ ] C. The code will not compile because of line 4.
- [ ] D. The code will not compile because of line 7.
- [ ] E. It compiles but throws an exception at runtime.

✗ 7. Which of the following statements about polymorphism are true? (Choose all that apply 0/1

- [x] A. A reference to an object may be cast to a subclass of the object without an explicit cast. ✗
- [ ] B. If a method takes a superclass of three objects, then any of those classes may be passed as a parameter to the method.
- [ ] C. A method that takes a parameter with type java.lang.Object will take any reference.
- [ ] D. All cast exceptions can be detected at compile-time.
- [x] E. By defining a public instance method in the superclass, you guarantee that the specific method will be called in the parent class at runtime. ✗

✗ 8. Choose the correct statement about the following code: 1: public interface Herbivore { 0/1
2: int amount = 10;
3: public static void eatGrass();
4: public int chew() { 5: return 13; 6: } 7: }

- A. It compiles and runs without issue.
- B. The code will not compile because of line 2.
- C. The code will not compile because of line 3. ✗
- D. The code will not compile because of line 4.
- E. The code will not compile because of lines 2 and 3.
- F. The code will not compile because of lines 3 and 4.

✓ 9. Choose the correct statement about the following code: 1: public interface CanFly { 1/1
2: void fly();
3: }
4: interface HasWings {
5: public abstract Object getWindSpan();
6: }
7: abstract class Falcon implements CanFly, HasWings { 8: }

- (●) A. It compiles without issue. ✓
- ( ) B. The code will not compile because of line 2.
- ( ) C. The code will not compile because of line 4.
- ( ) D. The code will not compile because of line 5.
- ( ) E. The code will not compile because of lines 2 and 5.
- ( ) F. The code will not compile because the class Falcon doesn't implement the interface
- ( ) methods.

✗ 10. Which statements are true for both abstract classes and interfaces? (Choose all that apply 0/1

- [x] A. All methods within them are assumed to be abstract. ✗
- [x] B. Both can contain public static final variables. ✓
- [ ] C. Both can be extended using the extend keyword.
- [ ] D. Both can contain default methods.
- [ ] E. Both can contain static methods.
- [ ] F. Neither can be instantiated directly.
- [x] G. Both inherit java.lang.Object. ✗

✓ 11. What modifiers are assumed for all interface variables? (Choose all that apply) 1/1

- [x] A. public ✓
- [ ] B. protected
- [ ] C. private
- [x] D. static ✓
- [x] E. final ✓
- [ ] F. abstract

✓ 12. What is the output of the following code? 1: interface Nocturnal { 1/1
2: default boolean isBlind() { return true; }
3: }
4: public class Owl implements Nocturnal {
5: public boolean isBlind() { return false; }
6: public static void main(String[] args) {
7: Nocturnal nocturnal = (Nocturnal)new Owl();
8: System.out.println(nocturnal.isBlind());
9: } 10: }

- A. true
- B. false ✓
- C. The code will not compile because of line 2.
- D. The code will not compile because of line 5.
- E. The code will not compile because of line 7.
- F. The code will not compile because of line 8

✗ 13. What is the output of the following code? 0/1

```
1: class Arthropod
2: public void printName(double input) { System.out .print("Arthropod"); }
3: } 4: public class Spider extends Arthropod {
5: public void printName(int input) { System.out.print("Spider"); }
6: public static void main(String[] args) {
7: Spider spider = new Spider();
8: spider.printName(4);
9: spider.printName(9.0); 10: } 11: }
```

- A. SpiderArthropod
- B. ArthropodSpider
- C. SpiderSpider
- D. ArthropodArthropod
- E. The code will not compile because of line 5.
- F. The code will not compile because of line 9. ✗

✓ 14. Which statements are true about the following code? (Choose all that apply) 1/1

```
1: interface HasVocalCords {
2: public abstract void makeSound();
3: }
4: public interface CanBark extends HasVocalCords { 5: public void bark();
6: }
```

- [ ] A. The CanBark interface doesn't compile.
- [ ] B. A class that implements HasVocalCords must override the makeSound() method.
- [x] C. A class that implements CanBark inherits both the makeSound() and bark() methods. ✓
- [ ] D. A class that implements CanBark only inherits the bark() method.
- [ ] E. An interface cannot extend another interface.

✗ 15. Which of the following is true about a concrete subclass? (Choose all that apply) 0/1

- [x] A. A concrete subclass can be declared as abstract. ✗
- [x] B. A concrete subclass must implement all inherited abstract methods. ✓
- [x] C. A concrete subclass must implement all methods defined in an inherited interface. ✗
- [ ] D. A concrete subclass cannot be marked as final.
- [ ] E. Abstract methods cannot be overridden by a concrete subclass.

✗ 16. What is the output of the following code? 1: abstract class Reptile { 0/1
2: public final void layEggs() { System.out.println("Reptile laying eggs"); }
3: public static void main(String[] args) {
4: Reptile reptile = new Lizard();
5: reptile.layEggs(); 6: }
7: }
8: public class Lizard extends Reptile {
9: public void layEggs() { System.out.println("Lizard laying eggs"); } 10: }

- ○ A. Reptile laying eggs
- ○ B. Lizard laying eggs
- ○ C. The code will not compile because of line 4.
- ◉ D. The code will not compile because of line 5. ✗
- ○ E. The code will not compile because of line 9.

✗ 17. What is the output of the following code? 0/1
1: public abstract class Whale {
2: public abstract void dive() {};
3: public static void main(String[] args) {
4: Whale whale = new Orca();
5: whale.dive(); 6: }
7: } 8: class Orca extends Whale {
9: public void dive(int depth) { System.out.println("Orca diving"); } 10: }

- ☐ A. Orca diving
- ☐ B. The code will not compile because of line 2.
- ☐ C. The code will not compile because of line 8.
- ☑ D. The code will not compile because of line 9. ✗
- ☐ E. The output cannot be determined from the code provided.

✗ 18. What is the output of the following code? (Choose all that apply) 1: 0/1

```
interface Aquatic {
2: public default int getNumberOfGills(int input) { return 2; } 3: }
4: public class ClownFish implements Aquatic {
5: public String getNumberOfGills() { return "4"; }
 6: public String getNumberOfGills(int input) { return "6"; }
7: public static void main(String[] args) {
8: System.out.println(new ClownFish().getNumberOfGills(-1));
9: } 10: }
```

- A. 2
- B. 4
- C. 6 ✗
- D. The code will not compile because of line 5.
- E. The code will not compile because of line 6.
- F. The code will not compile because of line 8.

✗ 19. Which of the following statements can be inserted in the blank so that the code will compile successfully? (Choose all that apply) 0/1

```
public class Snake {}
public class Cobra extends Snake {}
public class GardenSnake {}
public class SnakeHandler {
 private Snake snake;
 public void setSnake(Snake snake)
 { this.snake = snake; } public static void main(String[] args) {
 new SnakeHandler().setSnake( ); } }
```

- [x] A. new Cobra() ✓
- [ ] B. new GardenSnake()
- [x] C. new Snake() ✓
- [x] D. new Object() ✗
- [ ] E. new String("Snake")
- [ ] F. null

✗ 20. What is the result of the following code? 0/1

```
1: public abstract class Bird {
2: private void fly() { System.out.println("Bird is flying"); } 3: public static
void main(String[] args) {
4: Bird bird = new Pelican();
5: bird.fly();
6: } 7: }
8: class Pelican extends Bird {
9: protected void fly() { System.out.println("Pelican is flying"); } 10: }
```

- ○ A. Bird is flying
- ○ B. Pelican is flying
- ◉ C. The code will not compile because of line 4. ✗
- ○ D. The code will not compile because of line 5.
- ○ E. The code will not compile because of line 9.

✗ 21. Which of the following statements are true? (Choose all that apply 0/1

- ☐ A. Runtime exceptions are the same thing as checked exceptions.
- ☑ B. Runtime exceptions are the same thing as unchecked exceptions. ✓
- ☑ C. You can declare only checked exceptions. ✗
- ☐ D. You can declare only unchecked exceptions.
- ☐ E. You can handle only Exception subclasses.

✓ 2. Which of the following pairs fill in the blanks to make this code compile? (Choose all that apply) 1/1

7: public void ohNo() _____ Exception { 8: _____ Exception(); 9: }

- [ ] A. On line 7, fill in throw
- [x] B. On line 7, fill in throws ✓
- [ ] C. On line 8, fill in throw
- [x] D. On line 8, fill in throw new ✓
- [ ] E. On line 8, fill in throws
- [ ] F. On line 8, fill in throws new

✗ 23. When are you required to use a finally block in a regular try statement (not a try-withresources)? 0/1

- ( ) A. Never.
- (•) B. When the program code doesn't terminate on its own. ✗
- ( ) C. When there are no catch blocks in a try statement.
- ( ) D. When there is exactly one catch block in a try statement.
- ( ) E. When there are two or more catch blocks in a try statement.

✓ 2 4. Which exception will the following throw? Object obj = new Integer(3); String str = (String) obj; System.out.println(str); 1/1

- ( ) A. ArrayIndexOutOfBoundsException
- (•) B. ClassCastException ✓
- ( ) C. IllegalArgumentException
- ( ) D. NumberFormatException
- ( ) E. None of the above.

✗ 25. Which of the following exceptions are thrown by the JVM? (Choose all that apply) 0/1

- [ ] A. ArrayIndexOutOfBoundsException
- [ ] B. ExceptionInInitializerError
- [x] C. java.io.IOException ✗
- [ ] D. NullPointerException
- [ ] E. NumberFormatException

✓ 26. What will happen if you add the statement System.out.println(5 / 0); to a working main() method? 1/1

- A. It will not compile.
- B. It will not run.
- C. It will run and throw an ArithmeticException. ✓
- D. It will run and throw an IllegalArgumentException.
- E. None of the above.

✓ 27. What is printed besides the stack trace caused by the NullPointerException from line 16? 1/1

```
1: public class DoSomething {
2: public void go() {
 3: System.out.print("A");
4: try {
5: stop();
6: } catch (ArithmeticException e) {
7: System.out.print("B");
8: } finally {
9: System.out.print("C");
10: }
11: System.out.print("D");
12: }
13: public void stop() {
14: System.out.print("E");
15: Object x = null;
16: x.toString();
17: System.out.print("F");
18: }
 19: public static void main(String[] args) {
 20: new DoSomething().go();
21: } 22: }
```

- A. AE
- B. AEBCD
- C. AEC ✓
- D. AECD
- E. No output appears other than the stack trace.

✗ 2 8. What is the output of the following snippet, assuming a and b are both 0?

```
3: try {
4: return a / b;
5: } catch (RuntimeException e) {
6: return -1;
7: } catch (ArithmeticException e) {
8: return 0;
9: } finally {
10: System.out.print("done");
11: }
```

- A. -1
- B. 0
- C. done-1
- D. done0
- E. The code does not compile.
- F. An uncaught exception is thrown. ✗

✗ 29. What is the output of the following program? 0/1

```
1: public class Laptop {
2: public void start() {
3: try {
4: System.out.print("Starting up ");
5: throw new Exception();
6: } catch (Exception e) {
7: System.out.print("Problem ");
8: System.exit(0);
9: } finally {
10: System.out.print("Shutting down ");
11: } 12: }
13: public static void main(String[] args) {
14: new Laptop().start(); 15: } }
```

- [ ] A. Starting up
- [ ] B. Starting up Problem
- [ ] C. Starting up Problem Shutting down
- [ ] D. Starting up Shutting down
- [ ] E. The code does not compile.
- [ ] F. An uncaught exception is thrown.

✓ 30. What is the output of the following program? 1/1

```
1: public class Dog {
2: public String name;
3: public void parseName() {
 4: System.out.print("1");
5: try { 6: System.out.print("2");
7: int x = Integer.parseInt(name);
 8: System.out.print("3");
9: } catch (NumberFormatException e) {
10: System.out.print("4"); 11: } 12: }
13: public static void main(String[] args) {
14: Dog leroy = new Dog();
15: leroy.name = "Leroy";
16: leroy.parseName();
17: System.out.print("5"); 18: }
```

- [ ] A. 12
- [ ] B. 1234
- [ ] C. 1235
- [ ] D. 124
- [x] E. 1245 ✓
- [ ] F. The code does not compile.
- [ ] G. An uncaught exception is thrown.

✓ 31. What is the output of the following program? 1: public class Cat { 1/1
2: public String name;
3: public void parseName() {
4: System.out.print("1");
5: try {
6: System.out.print("2");
7: int x = Integer.parseInt(name);
8: System.out.print("3");
9: } catch (NullPointerException e) {
10: System.out.print("4");
11: } 12: System.out.print("5");
13: }
14: public static void main(String[] args) {
15: Cat leo = new Cat(); 16: leo.name = "Leo";
17: leo.parseName();
18: System.out.print("6"); 19: } 20: }

- [x] A. 12, followed by a stack trace for a NumberFormatException ✓
- [ ] B. 124, followed by a stack trace for a NumberFormatException
- [ ] C. 12456
- [ ] D. 12456
- [ ] E. 1256, followed by a stack trace for a NumberFormatException
- [ ] F. The code does not compile.
- [ ] G. An uncaught exception is thrown

✗ 32. What is printed by the following? (Choose all that apply) 1: public class Mouse { 0/1
2: public String name;
3: public void run() {
4: System.out.print("1");
5: try {
6: System.out.print("2");
7: name.toString();
8: System.out.print("3");
9: } catch (NullPointerException e) {
10: System.out.print("4");
11: throw e; 12: }
13: System.out.print("5"); 14: }
15: public static void main(String[] args) {
16: Mouse jerry = new Mouse();
17: jerry.run();
18: System.out.print("6"); 19: } }

- [ ] A. 1
- [ ] B. 2
- [ ] C. 3
- [ ] D. 4
- [ ] E. 5
- [ ] F. 6
- [x] G. The stack trace for a NullPointerException ✓

✗ 33. Which of the following statements are true? (Choose all that apply) 0/1

- [ ] A. You can declare a method with Exception as the return type.
- [ ] B. You can declare any subclass of Error in the throws part of a method declaration.
- [ ] C. You can declare any subclass of Exception in the throws part of a method declaration.
- [x] D. You can declare any subclass of Object in the throws part of a method declaration. ✗
- [x] E. You can declare any subclass of RuntimeException in the throws part of a method declaration. ✓
- [ ] Option 6

✗ 34. Which of the following can be inserted on line 8 to make this code compile? (Choose all that apply) 7: public void ohNo() throws IOException { 8: // INSERT CODE HERE 9: } 0/1

- [ ] A. System.out.println("it's ok");
- [x] B. throw new Exception(); ✗
- [ ] C. throw new IllegalArgumentException();
- [x] D. throw new java.io.IOException(); ✓
- [ ] E. throw new RuntimeException();

✗ 35. Which of the following are unchecked exceptions? (Choose all that apply) 0/1

- [x] A. ArrayIndexOutOfBoundsException ✓
- [ ] B. IllegalArgumentException
- [ ] C. IOException
- [x] D. NumberFormatException ✓
- [x] E. Any exception that extends RuntimeException ✓
- [ ] F. Any exception that extends Exception

✗ 36. Which scenario is the best use of an exception? 0/1

- [ ] A. An element is not found when searching a list.
- [x] B. An unexpected parameter is passed into a method. ✓
- [ ] C. The computer caught fire.
- [ ] D. You want to loop through a list.
- [x] E. You don't know how to code a method. ✗

✗ 37. Which of the following can be inserted into Lion to make this code compile? (Choose all that apply) 0/1

class HasSoreThroatException extends Exception {} class TiredException extends RuntimeException {} interface Roar { void roar() throws HasSoreThroatException; }
class Lion implements Roar {// INSERT CODE HERE }

- [ ] A. public void roar(){}
- [x] B. public void roar() throws Exception{} ✗
- [x] C. public void roar() throws HasSoreThroatException{} ✓
- [ ] D. public void roar() throws IllegalArgumentException{}
- [ ] E. public void roar() throws TiredException{}

✗ 38. Which of the following are true? (Choose all that apply 0/1

- [x] A. Checked exceptions are allowed to be handled or declared. ✓
- [x] B. Checked exceptions are required to be handled or declared. ✓
- [ ] C. Errors are allowed to be handled or declared.
- [x] D. Errors are required to be handled or declared. ✗
- [x] E. Runtime exceptions are allowed to be handled or declared. ✓
- [ ] F. Runtime exceptions are required to be handled or declared.

✗ 39. Which of the following can be inserted in the blank to make the code compile? (Choose all that apply) public static void main(String[] args) { 0/1

```
try {
System.out.println("work real hard");
}
catch ( e) { }
catch (RuntimeException e) { }
}
```

- [ ] A. Exception
- [ ] B. IOException
- [ ] C. IllegalArgumentException
- [ ] D. RuntimeException
- [ ] E. StackOverflowError
- [x] F. None of the above. ✗

✗ 40. What does the output of the following contain? (Choose all that apply) 0/1

```
12: public static void main(String[] args) {
 13: System.out.print("a");
14: try {
 15: System.out.print("b");
16: throw new IllegalArgumentException();
 17: } catch (IllegalArgumentException e) {
18: System.out.print("c");
19: throw new RuntimeException("1");
20: } catch (RuntimeException e) {
21: System.out.print("d");
22: throw new RuntimeException("2");
23: } finally {
24: System.out.print("e");
25: throw new RuntimeException("3");
26: } 27: }
```

- [x] A. abce ✓
- [ ] B. abde
- [ ] C. An exception with the message set to "1"
- [ ] D. An exception with the message set to "2"
- [ ] E. An exception with the message set to "3"
- [ ] F. Nothing; the code does not compile.