

Question 1

0 out of 3 points

Given the class hierarchy below, what output would be generated with the following statement:
Cyberman one = new Cyberman("Bob");

```
public class Robot
{
    public Robot() { System.out.println("Tom Servo"); }
    public Robot(String name) { System.out.println(name); }
}

public class Cyborg extends Robot
{
    public Cyborg() { System.out.println("Robocop"); }
    public Cyborg(String name) { System.out.println(name); }
}

public class Cyberman extends Cyborg
{
    public Cyberman() { super(); System.out.println("Cyberman"); }
    public Cyberman(String name) { super(name); System.out.println(name); }
}
```

Answer:

Tom Servo

Robocop

Cyberman

Pg 417

Constructors that do not specifically invoke a parent constructor will automatically call the default (no argument) constructor

Question 2

3 out of 5 points

Given the code below, what variables in the instantiated Doctor object can be accessed without mutators or accessors?

```
package DoctorWho;
class TimeTraveller
{
    static age;
    public String name;
    protected char gender;
}
```

```
package DoctorWho;
public class Doctor extends TimeTraveller
{
    private int number;
    protected String companion;
    String uniqueItem;
}
```

```
package DoctorWho;
public class TVShow
{
    Doctor Who = new Doctor;
}
```

Answers:

- **age**
- **name**
- **gender**
- number
- **companion**
- **uniqueItem**

Pg 441

The protected modifier allows access from any classes in the same package. So everything is able to be accessed except for the private variable number.

Question 5

0 out of 3 points

What is the value of TARDIS at the end of this code segment?

```
void TimeTravel (String Doctor)
{
    Doctor = "Who";
}
String TARDIS;
TimeTravel(TARDIS);
```

Answer:

- **empty string**

Pg 212

When passing a variable as an argument it passes the value of the variable, not the variable itself. It doesn't actually assign TARDIS = "Who"

Question 7

3.03 out of 6 points

Consider that the following variables have been declared:

String companion11 = "Clara";

String companion12 = "Clara";

Which of the following comparisons will return true? Select all that apply.

Answers:

- **companion11.equals(companion12)**
- companion11 == companion12
- companion11.greaterThanEqualTo(companion12)
- **companion11.contains("lar")**
- **companion11.compareTo(companion12)**
- **companion11.equalsIgnoreCase(companion12)**

Pg 133

Technically compareTo returns 0, but this is interpreted as true

Question 8

2 out of 4 points

```
int David_Tennant;  
double Matt_Smith;
```

Answers:

- **Matt_Smith = (double)David_Tennant;**
- Matt_Smith = David_Tennant;
 - //Remember this question is about syntax and not personal opinion
- Matt_Smith = double(David_Tennant;)
- (double)David_Tennant = Matt_Smith;
- **Integer tenth = David_Tennant;**
- **Double eleventh = 11.50;**
- David_Tennant = eleventh.intValue();

Pg 381

intValue() can only be used on Integer and Double objects, not the primitive data types

Question 9

1.02999 out of 6 points

```
public class SonicScrewdriver  
{  
int functions;  
public static void addFunction()  
{ functions += 1; }  
public static abstract doAnything();
```

```
@Override  
public void toString()  
{ System.out.println(functions); }  
}
```

Answers:

- **declaration of class variable**
- addFunction header
- adding one to variable
- **doAnything function header**
- @Override annotation
- **toString function header**

Pg 496

SonicScrewdriver is an abstract class because of doAnything method, and because of that it needs the abstract modifier in class declaration.

Question 15

1.5 out of 3 points

Which of the following is a valid call for the generic method declared below?

```
public class Whovian {  
    public static <E> void print(E[] list)  
    { //print array }  
    public static void main(String[] args) {  
        String [] Companions = {"Rose", "Amy", "Rory", "Clara", "Bill"};  
        //call generic function  
    }  
}
```

Answers:

- **<String>print(Companions);**
- **print(Companions);**
- **Whovian.<String>print(Companions);**
- **print(<String>Companions);**

Pg 743

You can explicitly tell the generic method what type of args you are using, but you don't have to

Question 16

3.02999 out of 6 points

Which of the following items regarding inheritance are true? Select all that apply.

Answers:

- the child class can change the access specifier for members of the parent class
- **child classes become abstract if abstract parent class methods are not defined in the child class**
- the child class can call the grandparent constructor (assuming a grandparent exists) by using the `super().super()` syntax
- **the child class can be in a different package from the parent class.**
- the child class can be generic even if the parent is not generic
- child classes can override but not overload parent class methods

Pg 416 - `super().super()` doesn't work.

Pg 420 - child classes CAN overload parent methods

Question 17

0.05999 out of 6 points

Which of the following statements are true about Scanners. Select all that apply.

Answers:

- **Multiple scanners can be used on the same file.**
- **The delimiter of a Scanner can be changed to something other than whitespace**
- **The Scanner object throws an InputMismatchException when incorrect input types are read**
- Scanners can be used to read a single character from the input stream
- Scanners are located in the java.io package
- Only one Scanner object is allowed in the same scope of a program

Pg 479 - scanner does throw InputMismatchException

Pg 478 - java.util.Scanner, not java.io

Question 18

3.03 out of 6 points

Which of the following statements represent a valid way to create a 2D array? Select all that apply.

Answers:

- `String [][] allCompanions = {"Susan", "Barbara"}, {"Polly", "Ben"};`
- **`String [][] allCompanions = {"Susan", "Barbara"};`**
- **`String [][] allCompanions = new String[13][];`**
- **`String [][] allCompanions = {"Susan", "Barbara", "Ian"}, {"Polly", "Ben"};`**
- `String [][] allCompanions = new String[100];`
- `String [][] allCompanions = new String[][11];`

Pg 290 - leaving second level array length blank is allowed, just not first level

Question 19

0 out of 3 points

Which of the following statements will produce the following output?

Dalek 34.79%

Answers:

- `System.out.printf("%s%-10f%", "Dalek", 34.79);`
- `System.out.printf("%10s%f%c", "Dalek", 34.79, "%");`
- `System.out.printf("%10s%f%c", "Dalek", 34.79, '%');`
- **`System.out.printf("%-10s%f%%", "Dalek", 34.79);`**

Pg 417 - use minus sign for "left justify" which will output "Dalek__" instead of "__Dalek"