Test_part2

1. Which option defines a well-encapsulated class?

```
a class Template {
public String font;
}
b class Template2 {
public String font;
public void setFont(String font) {
this.font = font;
}
public String getFont() {
return font;
}
}
c class Template3 {
private String font;
public String author;
public void setFont(String font) {
this.font = font;
}
public String getFont() {
return font;
}
public void setAuthor(String author) {
this.author = author;
}
public String getAuthor() {
return author;
}
}
```

d None of the above.

2. Examine the following code and select the correct option(s):

```
public class Person {
public int height;
public void setHeight(int newHeight) {
if (newHeight <= 300)
height = newHeight;
}
}
```

a The height of a Person can never be set to more than 300.
b The preceding code is an example of a well-encapsulated class.
c The class would be better encapsulated if the height validation weren't set to 300.
d Even though the class isn't well encapsulated, it can be inherited by other classes.

3. Which of the following methods correctly accepts three integers as method arguments and returns their sum as a floating-point number?

```
a public void addNumbers(byte arg1, int arg2, int arg3) {
double sum = arg1 + arg2 + arg3;
}
b public double subtractNumbers(byte arg1, int arg2, int arg3) {
double sum = arg1 + arg2 + arg3;
return sum;
}
c public double numbers(long arg1, byte arg2, double arg3) {
return arg1 + arg2 + arg3;
}
d public float wakaWakaAfrica(long a1, long a2, short a977) {
double sum = a1 + a2 + a977;
return (float)sum;
}
```

4. Which of the following statements are true?
a If the return type of a method is int, the method can return a value of type
byte.
b A method may or may not return a value.
c If the return type of a method is void, it can define a return statement without
a value, as follows:
return;
d A method may or may not accept any method arguments.
e A method should accept at least one method argument or define its return type.

f A method whose return type is String can’t return null.

5. Given the following definition of class `Person`,

```
class Person {
public String name;
public int height;
}
```

what is the output of the following code?

```
class EJavaGuruPassObjects1 {
public static void main(String args[]) {
Person p = new Person();
p.name = "EJava";
anotherMethod(p);
System.out.println(p.name);
someMethod(p);
System.out.println(p.name);
}
static void someMethod(Person p) {
p.name = "someMethod";
System.out.println(p.name);
}
static void anotherMethod(Person p) {
p = new Person();
p.name = "anotherMethod";
System.out.println(p.name);
}
}
```

a `anotherMethod`
```
anotherMethod
someMethod
someMethod
```
b `anotherMethod`
```
EJava
someMethod
someMethod
```
c `anotherMethod`
```
EJava
someMethod
EJava
```
d `Compilation error`

6. What is the output of the following code?

```
class EJavaGuruPassPrim {
public static void main(String args[]) {
int ejg = 10;
anotherMethod(ejg);
System.out.println(ejg);
someMethod(ejg);
System.out.println(ejg);
}

static void someMethod(int val) {
++val;
System.out.println(val);
}
static void anotherMethod(int val) {
val = 20;
System.out.println(val);
}
}
```

a `20`
```
10
11
11
```
b `20`
```
20
11
10
```
c `20`
```
10
11
10
```
d `Compilation error`

7. Given the following signature of method `eJava`, choose the options that correctly overload this method:

```
public String eJava(int age, String name, double duration)
```

a `private String eJava(int val, String firstName, double dur)`

b `public void eJava(int val1, String val2, double val3)`
c `String eJava(String name, int age, double duration)`
d `float eJava(double name, String age, byte duration)`
e `ArrayList<String> eJava()`
f `char[] eJava(double numbers)`
g `String eJava()`

8. Given the following code,

```
class Course {
void enroll(long duration) {
System.out.println("long");
}
void enroll(int duration) {
System.out.println("int");
}
void enroll(String s) {
System.out.println("String");
}
void enroll(Object o) {
System.out.println("Object");
}
}
```

what is the output of the following code?

```
class EJavaGuru {
public static void main(String args[]) {
Course course = new Course();
char c = 10;
course.enroll(c);
course.enroll("Object");
}
}
```

```
a Compilation error
b Runtime exception
c int
String
d long
Object
```

9. Examine the following code and select the correct options:

```
class EJava {
public EJava() {
this(7);
System.out.println("public");
}
private EJava(int val) {
this("Sunday");
System.out.println("private");
}
protected EJava(String val) {
System.out.println("protected");
}
}
class TestEJava {
public static void main(String[] args) {
EJava eJava = new EJava();
}
}
```

a The class `EJava` defines three overloaded constructors.
b The class `EJava` defines two overloaded constructors. The private constructor isn't counted as an overloaded constructor.
c Constructors with different access modifiers can't call each other.
d The code prints the following:

```
protected
private
public
```

e The code prints the following:

```
public
private
protected
```

10. Select the incorrect options:
a If a user defines a private constructor for a public class, Java creates a public default constructor for the class.
b A class that gets a default constructor doesn't have overloaded constructors.
c A user can overload the default constructor of a class.

d The following class is eligible for default constructor:
class EJava {}
e The following class is also eligible for a default constructor:

```
class EJava {
void EJava() {}
}
```

11. What is the output of the following code?

```
class Animal {
void jump() { System.out.println("Animal"); }
}
class Cat extends Animal {
void jump(int a) { System.out.println("Cat"); }
}
class Rabbit extends Animal {
void jump() { System.out.println("Rabbit"); }
}
class Circus {
public static void main(String args[]) {
Animal cat = new Cat();
Rabbit rabbit = new Rabbit();
cat.jump();
rabbit.jump();
}
}
```

a Animal
Rabbit
b Cat
Rabbit
c Animal
Animal
d None of the above

12. Given the following code, select the correct statements:

```
class Flower {
public void fragrance() {System.out.println("Flower"); }
}
class Rose {
public void fragrance() {System.out.println("Rose"); }
}
class Lily {
public void fragrance() {System.out.println("Lily"); }
}
class Bouquet {
public void arrangeFlowers() {
Flower f1 = new Rose();
Flower f2 = new Lily();
f1.fragrance();
}
}
```

a The output of the code is
Flower
b The output of the code is
Rose
c The output of the code is
Lily
d The code fails to compile.

13. Examine the following code and select the correct method declaration to be inserted at `//INSERT CODE HERE`:

```
interface Movable {
void move();
}
class Person implements Movable {
public void move() { System.out.println("Person move"); }
}
class Vehicle implements Movable {
public void move() { System.out.println("Vehicle move"); }
}
class Test {
// INSERT CODE HERE
movable.move();
}
}
```

a `void walk(Movable movable) {`
b `void walk(Person movable) {`

c `void walk(Vehicle movable) {`
d `void walk() {`

14. Select the correct statements:
a Only an `abstract` class can be used as a base class to implement polymorphism with classes.
b Polymorphic methods are also called overridden methods.
c In polymorphism, depending on the exact type of object, the JVM executes the appropriate method at compile time.
d None of the above.

15. Given the following code, select the correct statements:

```
class Person {}
class Employee extends Person {}
class Doctor extends Person {}
```

a The code exhibits polymorphism with classes.
b The code exhibits polymorphism with interfaces.
c The code exhibits polymorphism with classes and interfaces.
d None of the above.

16. Which of the following statements are true?
a Inheritance enables you to reuse existing code.
b Inheritance saves you from having to modify common code in multiple classes.
c Polymorphism passes special instructions to the compiler so that the code can run on multiple platforms.
d Polymorphic methods can't throw exceptions.

17. Given the following code, which of the options are true?

```
class Satellite {
void orbit() {}
}
class Moon extends Satellite {
void orbit() {}
}
class ArtificialSatellite extends Satellite {
void orbit() {}
}
```

a The method `orbit` defined in the classes `Satellite`, `Moon`, and `ArtificialSatellite` is polymorphic.
b Only the method `orbit` defined in the classes `Satellite` and `ArtificialSatellite` is polymorphic.
c Only the method `orbit` defined in the class `ArtificialSatellite` is polymorphic.
d None of the above.

18. Examine the following code:

```
class Programmer {
void print() {
System.out.println("Programmer ");
}
}
class Author extends Programmer {
void print() {
System.out.println("Author ");
}
}
class TestEJava {
Programmer a = new Programmer();
// INSERT CODE HERE
a.print();
b.print();
}
```

Which of the following lines of code can be individually inserted at `//INSERT CODE HERE` so that the output of the code is as follows?

```
Programmer
Author
```

a `Programmer b = new Programmer();`
b `Programmer b = new Author();`
c `Author b = new Author();`
d `Author b = new Programmer();`
e `Programmer b = ((Author)new Programmer());`
f `Author b = ((Author)new Programmer());`

19. Given the following code, which of the options, when applied individually, will make it compile successfully?

```
Line1> interface Employee {}
Line2> interface Printable extends Employee {
Line3> String print();
Line4> }
Line5> class Programmer {
Line6> String print() { return("Programmer "); }
Line7> }
Line8> class Author extends Programmer implements Printable, Employee {
Line9> String print() { return("Author "); }
Line10> }
```

a Modify the code on line 2 to `interface Printable {`

b Modify the code on line 3 to public String print();

c Define the accessibility of the print methods to public on lines 6 and 9.

d Modify the code on line 8 so that it implements only the interface `Printable`.

20. What is the output of the following code?

```
class Base {
String var = "EJava";
void printVar() {
System.out.println(var);
}
}
class Derived extends Base {
String var = "Guru";
void printVar() {
System.out.println(var);
}
}
class QReference {
public static void main(String[] args) { Base base = new Base();
Base derived = new Derived();
System.out.println(base.var);
System.out.println(derived.var);
base.printVar();
derived.printVar();
}
}
```

a
```
EJava
EJava
EJava
Guru
```

b
```
EJava
Guru
EJava
Guru
```

c
```
EJava
EJava
EJava
EJava
```

d
```
EJava
Guru
Guru
Guru
```

21. The class Phone defines a local variable and an instance variable, phoneNumber, with the same name. Examine the definition of the method setNumber. Execute the class on your system and select the correct output of the class TestPhone from the given options:

```
class Phone {
String phoneNumber = "123456789";
void setNumber () {
String phoneNumber;
phoneNumber = "987654321";
}
}
class TestPhone {
public static void main(String[] args) {
Phone p1 = new Phone();
p1.setNumber();
System.out.println (p1.phoneNumber);
}
}
```

a 123456789

b 987654321
c No output
d The class Phone will not compile.