Theory

Q. What is method overloading in Java?

If you have two methods which do the same thing it's better they have the same name, but two methods cannot have the same name until you overload them. So overloading is a process of declaring two methods with the same name but different method signature like System.out which is object of PrintStream class has several println() method to print different data types e.g. byte, short, int, char, float and double. All of them are called the overloaded method. Overloaded method calls are resolved during compile time in Java and they must have different method signatures. See here to learn more about method overloading in Java

Some mcq question

1. What is the process of defining two or more methods within same class that have same name but different parameters declaration?

a) method overloading

b) method overriding

c) method hiding

d) none of the mentioned

ans (a)

2. Which of these can be overloaded?

a) Methods

b) Constructors

c) All of the mentioned

d) None of the mentioned

ans (c)

Edit with WPS Office

3. Which of these is correct about passing an argument by call-by-value process?

a) Copy of argument is made into the formal parameter of the subroutine

b) Reference to original argument is passed to formal parameter of the subroutine

c) Copy of argument is made into the formal parameter of the subroutine and changes made on parameters of subroutine have effect on original argument

d) Reference to original argument is passed to formal parameter of the subroutine and changes made on parameters of subroutine have effect on original argument

ans (a)

4. What is the process of defining a method in terms of itself, that is a method that calls itself?

a) Polymorphism

b) Abstraction

c) Encapsulation

d) Recursion

ans (d)

5. To successfully overload a method in Java, the return types must be ___.

A) Same

B) Different

C) Same but using superclass or subclass types also work

D) None

ans (C)

Edit with WPS Office

6. To successfully overload a method in Java, the argument-list or parameter-list must be ___.

A) Same

B) Different

C) -

D) -

ans (B)

7. What is the output of the below Java program with multiple methods?

```
public class MethodOverloading1
{
   void show(int a, char b)
   {
      System.out.println("KING KONG");
   }
   void show(char a, int b)
   {
      System.out.println("JIM JAM");
   }
            public static void main(String[] args)
   {
      MethodOverloading1 m = new MethodOverloading1();
      m.show(10, 'A');
      m.show('B', 10);
   }
}
```

A)

JIM JAM

JIM JAM

Edit with WPS Office

B)

KING KONG

KING KONG

C)

KING KONG

JIM JAM

D) compiler error

ans (c)

8. To successfully overload a method in Java, the method names must be ___.

A) Same

B) Different

C) Same or different

D) None

ans (A)

9. What is the output of the below Java program?

```
public class MethodOverloading2
{
   int info()
   {
      System.out.println("PLANE");
      return 0;
   }
   void info()
   {
      System.out.println("AIRPORT");
   }
```

Edit with WPS Office

```
    public static void main(String[] args)
    {
        MethodOverloading2 m = new MethodOverloading2();
        int a = m.info();
    }
}
```

A) PLANE

B) AIRPORT

C) Compiler error

D) None

ans (C)

10. What is the output of the below Java program with method overloading?

```
class Wood{ }
class SubWood extends Wood{ }

public class MethodOverloading3
{
    Wood display(int a)
    {
        System.out.println("PINE");
        return new Wood();
    }
    SubWood display()
    {
        System.out.println("TEAK");
        return new SubWood();
    }
```

Edit with WPS Office

```
    public static void main(String[] args)
    {
        MethodOverloading3 m = new MethodOverloading3();
        m.display();
    }
}
```

A) PINE

B) TEAK

C) Compiler error

D) None

ans (B)

11. What is the output of the below Java program trying to overload a method "jump"?

```
class Rabbit{ }
class WildRabbit extends Rabbit{ }

public class MethodOverloading4
{
    Rabbit jump()
    {
        System.out.println("Rabbit Jump");
        return new Rabbit();
    }
    WildRabbit jump()
    {
        System.out.println("Wild Rabbit Jump");
        return new WildRabbit();
```

Edit with WPS Office

```
    }

    public static void main(String[] args)
    {
        MethodOverloading4 obj = new MethodOverloading4();
        obj.jump();
    }
}
```

A) Rabbit Jump

B) Wild Rabbit Jump

C) Compiler error

D) None

ans (C)

12. Java method overloading implements the OOPS concept ___.

A) Inheritance

B) Polymorphism

C) Encapsulation

D) None

ans (B)

13. Which is the overloaded static method of Math class to get absolute value in Java?

A) Math.abs(int)

B) Math.abs(float)

C) Math.abs(double)

D) All the above

ans(D)

Edit with WPS Office

14. Method signature consists of

a) Method Name, Return Type and Number Of Arguments

b) Access Modifier, Method Name and Types Of Arguments

c) Method Name, Number Of Arguments, Types Of Arguments and Order Of Arguments

d) Return Type, Access Modifier and Order Of Arguments

ans (c)

15. In the below Class X, is method properly overloaded?

```
class X
{
	int method(int i, int d)
	{
		return i+d;
	}
	static int method(int i, double d)
	{
		return (int)(i+d);
	}
	double method(double i, int d)
	{
		return i+d;
	}
	static double method(double i, double d)
	{
		return i+d;
	}
}
```

a) Yes .

Edit with WPS Office

b) No .

c) Non of these.

ans (a)

16. What will be the output of the following program?

```
class X
{
	void method(int a)
	{
		System.out.println("ONE");
	}
	void method(double d)
	{
		System.out.println("TWO");
	}
}
class Y extends X
{
	@Override
	void method(double d)
	{
		System.out.println("THREE");
	}
}
public class MainClass
{
    public static void main(String[] args)
	{
		new Y().method(100);
	}
```

Edit with WPS Office

}

a) One.

b) Tan.

c) Hundred.

d) Five.

ans(a)

17. Can overloaded methods be synchronized?

a) Yes.

b) No.

c) All of these .

d) Non of these.

ans (a)

18. Why we use method overloading?

a) Increases the readability.

b) Decrease the readability.

c) Non of these.

ans (a)

19. How many ways we can achieve method overloading?

a) By changing number of arguments.

b) By changing the data types.

c) All of these.

d) Non of these.

ans (c)

20. What are the others name for java method overloading?

a) Compile-time polymorphism.

Edit with WPS Office

b) Static polymorphism.

c) Static binding .

d) All of the mention.

ans (d)

21. Can we achieve method overloading by changing the return -type?

a) No, we cannot achieve method overloading by changing the return type.

b) Yes, we can achieve method overloading by changing the return type.

c) Yes, It is possible overloaded method can be synchronized.

d) Non of these

ans (a)

22. Can we declare overloaded method as final?

a) Yes

b) No

c) All of these

ans(a)

23. Can not we overload constructor in java?

a) Yes

b) No

ans(b)

Edit with WPS Office