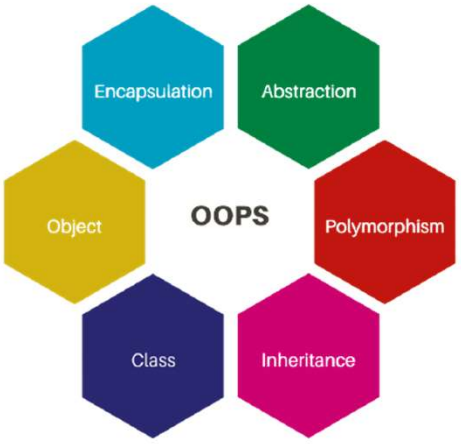


ABESIT
COLLEGE OF ENGINEERING
CRC-Training

Java™

Command Line Arguments , Revisiting Main , Method overloading

ABESIT
COLLEGE OF ENGINEERING
CRC-Training

Command Line Arguments

To pass an argument to the main method

Main method is invoked by the Java runtime system before the class or [illegible]ated

```
public static void main(String [] args){ }
```

Why is the argument mentioned as String ?

The String[] (array of **String** objects) contains the command line arguments that are passed ,when the program is invoked

Primitive values will not be able to hold the data that strings can

String arguments can easily be converted or parsed to primitive values with the help of some predefined methods

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Command Line Example 1

```
public class Main
{
    public static void main(String[] a
        String a = args[0];
        String b = args[1];

        System.out.println(a + " "+ b);
    }
}
```

Note how command line arguments are used

Note how args[1] is assigned to variable b

Output:

```
Hello Java
```

Command line arguments: Hello Java

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Command Line Example 2

```
public class Main
{
    public static void main(String[] args) {
        int a = Integer.parseInt(args[0]);
        int b = Integer.parseInt(args[1]);

        System.out.println(a + b);
    }
}
```

Note how args[0] and args[1] is converted to integer before assigning it to variables

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Revisiting main() method

main() is declared as public so it can be accessed from anywhere

static allows main( ) to be called without having to instantiate a particular instance of the class

This is required because the main( ) is called by the Java interpreter before any objects are created

void informs the compiler that main( ) does not return any value

The parameter passed to the main() i.e. String args[] facilitates input through command line arguments

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Method Overloading

- If a class has multiple methods having same name but different in parameters, it is known as Method Overloading.
- Why Method Overloading?
  - It is used when different objects are required to perform a similar set of tasks but use different input parameters.
- Best Example - println() method.
  - System.out.println(),
  - System.out.println(int)
  - System.out.println(double)
  - System.out.println(string)
  - System.out.println(character)

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Two ways to overload the method

- By changing the number of arguments
- By changing the data type

```
class Adder{
        int add(int a,int b){
                return a+b;
        }
        int add(int a,int b,int c){
                return a+b+c;
        }
}
```

```
class Adder{
        int add(int a,int b){
                return a+b;
        }
        double add(double a, double b) {
                return a+b;
        }
}
```


ABESIT COLLEGE OF ENGINEERING
CRC-Training

Can we overload java main() method?

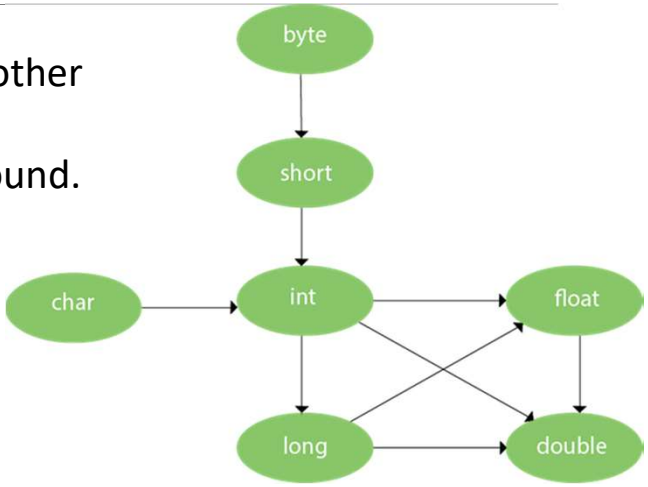
- Yes, by method overloading.
- But JVM calls main() method which receives string array as arguments only.

- **public static void** main(String[] args)
- **public static void** main(String args)
- **public static void** main(int args)

Method Overloading and Type Promotion

ABESIT COLLEGE OF ENGINEERING
CRC-Training

One type is promoted to another implicitly,
if no matching datatype is found.



ABESIT COLLEGE OF ENGINEERING
CRC-Training

Example - Overloading with TypePromotion

```
class cal{
    int add(int a, int b,int c){return a+b+c;}
    double add(int a, double b){return a+b;}
}
public class Main
{
    public static void main(String[] args) {
        cal obj1 = new cal();
        cal obj2 = new cal();
        System.out.println(obj1.add(10,20));
        System.out.println(obj1.add(10,20.5));
    }
}
```

now second int literal will be promoted to double

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Type Promotion in case of ambiguity

```
class cal{
    double add(double a, int b){return a+b;}
    double add(int a, double b){return a+b;}
}
public class Main
{
    public static void main(String[] args) {
        cal obj1 = new cal();
        System.out.println(obj1.add(10,20.5));
        System.out.println(obj1.add(10.5,20));
    }
}
```

```
30.5
30.5
```

What will happen when we call add a method like –

obj1.add(10,10);

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Type Promotion in case of ambiguity

```
1  class cal{
2      double add(double a, int b){return a+b;}
3      double add(int a, double b){return a+b;}
4  }
5  public class Main
6  {
7      public static void main(String[] args) {
8          cal obj1 = new cal();
9          System.out.println(obj1.add(10,10));
10     }
11 }
```

Compilation failed due to following error(s).

```
Main.java:9: error: reference to add is ambiguous
                System.out.println(obj1.add(10,10));
                                   ^
  both method add(double,int) in cal and method add(int,double) in cal match
1 error
```