

MCQ Questions on class and method with variable arguments

Question1 : what will be the output of given code?

```
public class VarargsExample
{
    public static void displayNames(String... names)
    {
        for (String mynames:names)
        {
            System.out.print(mynames + " ");
        }
    }
    public static void main(String args[])
    {
        displayNames("Alex","Richard","John");
    }
}
```

What will be the output after compiling and executing the preceding program?

- A. The program leads to compilation error.
- B. The program compiles successfully and displays “Alex Richard John” as output.
- C. The program compiles successfully and leads to runtime exception.
- D. The program compiles successfully but does not display anything as output.

The correct answer is B.

Explanation: The preceding program displays “Alex Richard John” as output as it demonstrates the variable argument list concept. Therefore, option B is correct

Question2: what will be the output of given code?

```
class Ques2 {
    int eval(int[]...vars)
    {
        int sum=0, b, c;
        for(b = 0; b<vars.length; b++) {
            for(c=0;c<vars[b].length; c++) {
                sum += vars[b][c];
            }
        }
        return(sum);
    }
    public static void main(String args[])
    {
        Ques2 varargs = new Ques2();
    }
}
```

```
int sum =0;
sum = varargs.eval(new int[]{10,20,30,40}, new int[]{40,50,60});
System.out.println("The sum of the numbers is:" + sum);
}
}
```

What will happen during compilation and execution of your program?

- A. The program will compile and display “The sum of the numbers is: 250” as output.
- B. The program will compile and display 25 as output.
- C. The program will not compile due to invalid declaration of integer variable arguments.
- D. The program will generate the runtime exception.

The correct option is A.

Explanation: The code will compile and execute successfully displaying “The sum of the numbers is: 250” as output because the eval (int []...vars) method is declared with a variable argument list as its parameter. In the code, a two dimensional array is declared to evaluate the sum and the eval method is invoked from the main method.

Question3 (what will be the output of given code)

```
class Ques3{
public static void main(String args[]) {
int x = 201;
myMethod(x++);
System.out.println(x);
}
static void myMethod(int x)
{   x %= 10;
System.out.println(x);
} }
```

What will be output of the above program after compilation and execution?

- A. The program will compile successfully and execute displaying 1 and 202 as output.
- B. The program will compile successfully and execute displaying 2 and 202 as output.

- C. The program will compile successfully and execute displaying 1 and 201 as output.
- D. The program will compile successfully and execute displaying 1 and 1 as output.

The correct option is A.

Explanation: After compiling and executing the above program 1 and 202 will be displayed as output. In the given program, the Ques12 class contains two static methods, the main () and myMethod (). Each of the static method defines a local variable x, having same name. When the program executes the myMethod () method is invoked and the value, 201 is passed as an argument to the myMethod () method which is assigned to its local variable, x. Then the compound operator performs the modulus operation and the resultant value (1) is displayed. Finally the value of the local variable within the main () method, i.e. 202 (after increment) is displayed.

Question 4

```
public class Ques4
{   public String name;
}
```

Now you realized that to make the name variable as read only for the other classes. Which of the following options are correct to mark the name variable as read only?

- A. You can mark the name variable as private.
- B. You can mark the name variable as private and provide the public method getName() which will return its value
- C. You can mark the name variable as protected.
- D. You can mark the name variable as static and provide the public static method getName() which will return its value

The correct option is B.

Explanation: In Java, the standard way to provide the read only access to the Variables are to mark the variable as private and provide a public method returning its value.

Question5: Which of the following statements are true based on the use of modifiers?

- A. Local variables can be declared either static or transient.
- B. The visibility of the local variables cannot be specified.
- C. By default the variable is accessible within the same class and subclass of the super class.
- D. The visibility of the local variables is default.

The correct option is B.

Explanation: The local variables cannot be marked as transient, volatile, and static, Correct option is B and The local variable does not have any accessibility as they are accessible only from the block in which they are declared.

Question6: Which of the following are valid declarations of the main () method?

- A. static main(String args[]){ }
- B. public static String main(String args[]) {... }
- C. public static void main(String args[]) {....}
- D. final static void main(String args[]) {....}

The correct option is C

Explanation: The following is a valid declaration of the static method:

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

Question7: Which of the following is the correct higher to lower order of restrictiveness for access specifiers?

- A. public> default(within the package)> protected> private
- B. private> default(within the package)> protected> public
- C. private> protected> default(within the package)> public
- D. protected> default(within the package)> private> public

The correct option is B.

Explanation: The private class members can be accessed only within the class in which they are declared and therefore the private access specifier is highly restrictive. Moreover, the members with default accessibility are accessible within the class in which they are declared and by the classes belonging to the

Same package. In addition, the protected members are also accessible from subclasses and therefore the protected access specifier is less restrictive as compared to the default accessibility.

Question 8: Imagine you want to clear your concept of nested classes and so you create a program containing nested and static classes. Consider that you have created the following program?

```
public class Ques8
{
    public static void main(String args[])
    {
        TestOuter o = new TestOuter();
        TestOuter.TestInner i = o.new TestInner();
        TestOuter.TestStaticInner inner = new TestOuter.TestStaticInner();
    }
}

class TestOuter {
    static int num1 = 100;
    TestOuter() {
        System.out.print("Welcome to the outer class" + " ");
    }
    class TestInner
    {
        TestInner() {
            System.out.print(TestOuter.num1 + " ");
        }
    }
    static class TestStaticInner {
        static int staticnum = 200;
        TestStaticInner() {
            System.out.print(staticnum + " ");
        }
    }
}
```

What will be the output after you compile and execute the preceding program?

- A. The program compiles successfully and displays “Welcome to the outer class 100 200” as output.
- B. The program compiles successfully and displays “Welcome to the outer class 200 100” as output.
- C. The program compiles successfully and displays “Welcome to the outer class 100” as output.
- D. The program compiles successfully and displays “Welcome to the outer class 200” as output.

The correct option is A.

Explanation: The first statement in the main method creates an instance of the TestOuter class and then the second statement instantiates the TestInner class. The instantiation of the TestInner class is done by using the outer class instance as the TestInner class is a non-static class. Finally an instance of the TestStaticInner class is created without using the instance of the TestOuter class as the TestStaticInner class is a static class. As a result the correct option is A.

Question9: Imagine that you are a Java programmer in the ABC Company and create the following program?

```
public class Ques9
{
    public void myMethod1()
    {
        static int num1=100;
        final int num2=200;
        System.out.println("The value of first variable is " + num1);
        System.out.println("The value of second variable is " + num2);
    }
    public void myMethod2()
    {
        int arr[] = new int[2];
        System.out.println(arr[arr.length-1]);
    }
    public static void main(String args[]) {
        new Ques9().myMethod1();
        new Ques9().myMethod2();
    }
}
```

```

}
}

```

What will be the output after you compile and execute the preceding program?

- A. The program will lead to compilation errors as static variables cannot be declared within methods.
- B. The program will compile successfully and display “The value of first variable is 100” and “The value of second variable is 200”, as output.
- C. The program will compile successfully and lead to the `ArrayIndexOutOfBoundsException` exception during runtime.
- D. The program will lead to compilation errors as the object `arr` is not initialized.

The correct option is A.

Explanation: The static variables cannot be declared within a method and therefore the correct option is A. However the final variables can be declared within a method and all array elements in an integer array are by default initialized to 0. As a result the correct option is A.

Question10: what will be the output of given code?

```

public class Ques10
{ private static int num1 = 100;
  private int num2 = 200;
  public static void myMethod1()
  {   num1 = 300;
      num2 = 400;
      System.out.println(num1 + “,” + num2);
  }
  public static void myMethod2()
  {
    num1 = 300;
    Ques10.num2 = 400;
  }
  public void myMethod3()
  {
    num1 = 300;
    num2 = 400;
  }
}

```

```

public void myMethod4()
{
    Ques10.num1 = 300;
    num2 = 400;
}
public static void main(String args[])
{
    Ques10 q = new Ques10();
    q.myMethod1();
}
}

```

Now you need to analyze the preceding program and give a feedback to your mentor with explanation. Therefore, which of the following statements you can provide as a feedback to your mentee?

- A. The program will compile successfully.
- B. The program will lead to compilation error as the non-static variables cannot be referenced from a static context.
- C. The program will compile successfully and lead to runtime error.
- D. The program will compile successfully and display “300,400” as output

The correct option is B

Explanation: The correct answer is B as it is not possible to access the non-static variable within the static method. Therefore options A, C, and D are incorrect.

Question 11: Imagine while practicing the concept of primitive variables in Java, you came across the following program ?

```

public class Ques11 {
    public static void main(String args[])
    {   Ques11 q = new Ques11();
        q.method(30);
        byte b = 3;
        q.method(b);
    }
    public void method(Integer i)

```



```

{ System.out.print("Integer value is: " + i + " ");
}
public void method(short s)
{ System.out.print("Short value is: " + s + " ");
}
public void method(byte t)
{ System.out.print("Byte value is: " + t + " ");
}
public void method(int num)
{ System.out.print("Int value is: " + num + " ");
}
}

```

What will be output of the preceding program?

- A. The program will display “Int value is: 30 Byte value is: 3” as output.
- B. The program will display “Integer value is: 30 Byte value is: 3” as output.
- C. The program will display “Int value is: 30 Short value is: 3” as output.
- D. The program will display “Integer value is: 30 Short value is: 3” as output

The correct option is A.

Explanation: In the preceding program, widening is preferred over boxing and therefore while invoking the method () method through the argument 30, it will be widened to call the method (int num).

Question12: Imagine you are working in the ABC Company and you are assigned a project with a team. Being a team leader you need to analyze the programs created by your team members. While analyzing the programs, you came across the following program:

```

public class Ques12
{
public static void main(String args[])
{
Ques12 q = new Ques12();
q.myMethod (10,20);
q.myMethod (new long[]{});
q.myMethod (new int[] {10,20});
}
}

```

```

void myMethod (short s1, short s2)
{
System.out.println ("short");
}
void myMethod (int i1, int i2)
{ System.out.println ("int");
}
void myMethod (int ...args)
{
System.out.println ("intargs");
}
}

```

Which of the following statements are justified in the context of the preceding program?

- A. The program will compile successfully and display “int intargs intargs” as output.
- B. The program will lead to compilation error.
- C. The program will compile successfully but lead to runtime exception.
- D. The program will display “short intargs intargs” as output.

The correct option is B.

Explanation: The program leads to compilation errors as while invoking the `q.myMethod(new long[]{})` method, implementation of the `myMethod()` method does not have the long array type argument. Therefore the option B is the correct answer.

Question13: Imagine you write the following program while understanding the concept of primitive variables?

```

public class Ques13
{ public static void main(String args[])
{
System.out.println(myMethod(myMethod(new int[] {10,20}),myMethod(10,20)));
}
static int myMethod(int num1, int num2)
{ return 10;
}
}

```

```
static int myMethod(int... args)
{ return 20;
}
}
```

What will be output of the preceding program?

- A. The program will compile successfully and display 10 as output.
- B. The program will lead to compile time error as the myMethod with int [], int [] argument is not defined.
- C. The program will compile successfully but lead to runtime exception.
- D. The program will compile successfully and display 20 as output

The correct option is A.

Explanation: In the preceding program, the code myMethod(new int[] {10,20}) invokes the myMethod(int...args) which returns 20. Moreover the code myMethod(10,20) invokes the myMethod(int num1, int num2) method which returns 10. Finally the myMethod(myMethod(new int[] {10,20}), myMethod(10,20)) method becomes myMethod(20,10) which invokes myMethod(int num1, int num2), thereby returning 10, which is displayed. Therefore the correct answer is A.

Question14: Imagine you are a Java programmer and you have created the following program?

```
public class Ques14 {
public static void main(String[] args)
{
System.out.println (myMethod (new double[] {10, 20, 30}));
System.out.println (myMethod (new Double[] {10d, 20d, 30d}));
System.out.println(myMethod(10, 20, 30));
System.out.println(myMethod());
}
static double myMethod(double ... args)
{
double total = 0;
for (double temp : args) {
total += temp;
}
```

```

}
return total;
}
static double myMethod(Double ... args)
{ double total = 2;
for (double temp : args) {
total *= temp;
}
return total;
}
}

```

What will be output of the preceding program?

- A. The program will lead to compilation error.
- B. The program will compile successfully and display “60.0 12000.0 60.0” as output.
- C. The program will compile successfully but lead to runtime error.
- D. The program will compile successfully and display “60.0 60.0 12000.0” as output.

The correct option is A.

Explanation: The program will lead to compilation error while invoking the myMethod() method without any argument. This is because of the ambiguity caused due to declarations of the myMethod taking primitive double variable argument and no declaration of the MyMethod without any argument.

Question15: Imagine being a Java programmer you write the following program

```

public class Ques15
{ String str;
int i=10;
static void myMethod() {
System.out.println("The value of String variable is" + new Ques15().str.length());
}
public static void main(String args[]) {
myMethod();
}
}

```

Which of the following statements are true in the context of the preceding program?

- A. The program will lead to compilation error as a non–static variable cannot be accessed from static context.
- B. The program will compile successfully but lead to runtime exception.
- C. The program will lead to compile time error as the String variable str is not assigned a value.
- D. The program will compile successfully and print 4 as output.

The correct option is B.

Explanation: The String variable str is not assigned a value so the default value null will be assigned. The myMethod() is a static method and the str variable is a non static variable, therefore the Ques15 class reference is used to access the str variable. The compiler will compile the program successfully as the default value null will be assigned to the str variable. However during execution the NullPointerException exception occurs as the length function is invoked on the str variable which is not initialized.

Question16: Imagine you are a faculty in an institute and you have explained the concept of Inner classes to the students. While practicing the students created the following program and you were asked to analyze the program?

```
public class Ques16
{ void myMethod()
{ System.out.println("Welcome to the world of programming");
}
}
class MyNest
{ public static void main(String args[]) {
Ques16 q = new Ques16();
q.myMethod();
}
}
}
```

What will be output of the preceding program?

- A. The program will compile successfully and print “Welcome to the world of programming” as output.
- B. The program will compile successfully but lead to runtime error.
- C. The program will lead to compilation error.

D. The program will compile successfully but no output is displayed

The correct option is C.

Explanation: The inner classes cannot have static declarations and so the preceding program will lead to compilation error.

Therefore the correct option is C.

Question17: Sam works in Xyz Company as Java programmer and he designed the following program?

```
class Rose
{
public void sam()
{
int y[] = {4, 2, 8};
for (int x=2; x<1+3*2-4; x++){
System.out.print(x+" ");
for (int j:y) {
j=j*x-4;
System.out.print(j+" ");
}
}
}
public static void main(String[] args)
{
Rose r = new Rose();
r.sam();
}
}
```

What would be the output of this program? Choose the correct option from the following:

- A. The program displays 2 4 2 8 B. The program displays 2 4 0 12
C. The program displays 2 4 4 16 D. The program displays 3 4 0 12

Correct option is B.

Explanation: Correct option is B. Firstly the value of the variable x is 2, which is less than the 3 (value specified in expression) therefore control will transfer into

for each loop and prints the array variable with modifications according to the expression. Array element is fetched into j. First array element is 4 and the Expression is $j \times x - 4$ i.e. $4 \times 2 - 4 = 4$. In the same way, other array elements are Extracted and displayed with modifications.

Question18: what will be the output of given code?

```
public class Rose
{
    protected void get(boolean x )
    {
        if(x)
        { System.out.println("True");
        }
        else
        { System.out.println("False");
        }
    }
    public static void main(String[] args)
    {
        Rose r = new Rose();
        r.get(true);
    }
}
```

What would be the output when the program is compiled?

- A. Program will display True
- B. Program will display False
- C. Program will successfully compile but give runtime error
- D. Program will not compile successfully

Option A is the correct answer

Explanation: Option A is the correct answer because the value true is passed in the get () method from main method. In the get () method, the if expression evaluates to true and statement written in it is displayed. Therefore, the Option B is incorrect. Option C and D are incorrect because the program will successfully compile and execute.

Question19: what will be the output of given code?

```

public class Rose
{
protected void get(char x )
{
switch(x)
{ case 88: System.out.println( "X");break;
case 90: System.out.println( "Z");break;
case 89: System.out.println( "Y");break;
default: System.out.println( 0);break;
case -97: System.out.println("a");break;
}
}
public static void main(String[] args)
{
Rose r = new Rose();
r.get('X');
}
}

```

What would be the output of this program?

- A. Program will display X B. Program will display X0
 C. Program will not compile successfully D. Program will compile successfully but not execute

Option C is the correct answer.

Explanation: Option C is the correct answer because switch case is designed to accept character values but - 97 is an integer value. Therefore, options A, B, and D are incorrect.

Question20: Sam as a developer in GIRI'S TECH HUB created the following program:

```

class Rose {
static int j;
public int arr()
{
int y[] = { 5 , 7, 8 , 6};
j = y[2]; return j;
}
}

```



```

}
public static void main (String args[])
{
    Rose r=new Rose();
    int x = r.arr( );
    System.out.println(x);
    switch(x)
    {
        case 0: System.out.print(0 + " ");break;
        case 2: System.out.print(2 + " ");break;
        case 8: System.out.print(8 + " ");
        case 5: System.out.print(5 + " ");break;
        default: System.out.print("Default");
    }
}
}
}

```

What would be the output when Sam compile and execute this program?

- A. Program will not compile successfully B. Program will display 8 8 5
 C. Program will display 8 5 D. Program will display 8

Option B is the correct answer.

Explanation: Option B is the correct answer. Firstly the control will transfer to arr () method and retrieves the array element at array [2] index and returns that value to calling routine. In the calling routine the retrieved value is first displayed and then used in switch statement. Therefore, the result is 8 8 and 5 is displayed because the case 8 does not have break and 5 is the statement written in next case. Therefore, options C and D are incorrect. Option A is incorrect because program will successfully compile.

MCQ Questions (static variables)

Question1: What will be the output of given code?

```

public class Time {
    int a = 50;
    int b = 10;
    public static void main (String args[]) {

```

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

Options

- a) 60
- b) 50
- c) 59
- e) Compilation Error

Ans: e (compilation Error – non static variable a cannot be reference from static context)

Explanation: we cannot use the non static variable in static function means as per our example we have the two variables name as a and b these declared as instance variable or non static and we have the function name as public static void main (String args []) this is the static function and we cannot use the non static variable a and b in static function so this is the major reason compiler generate the error to us non static variable a cannot reference from static context.

Question2: What will be the output of given code?

```
class W  
{ static int c = 0;  
public static void main(String[] args)  
{ W w1 = c();  
W w2 = c(w1);  
W w3 = c(w2);  
W w4 = c(w3);  
}  
private W()  
{  
System.out.println("c = " + c);  
}  
static W c()
```

```

{ return c++ <= 0 ? new W() : null;
}
static W c(W w)
{ return w.c++ == 1 ? new W() : null;
}
}

```

Options

- a) c = 1
c = 2
- b) c = 1
c = 2
c = 3
- c) c = 1
c = 2
c = 3
c = 4
- d) Compilation Error

Ans: option a

C=1

C=2

Explanation: if we think about the above code we have the one private default constructor name as W () and we have the static overloaded method name as C () which return the reference of W class.

If we think about the main () method in main method we have the statement W w1= C (); means we call the C() method version which has no parameter and in this method we have the statement return c++ <=0 ?new W():null here c++ is post increment so post increment having less priority than <= means here <= get executed before ++ means variable c compare before increment so the statement like as 0 <=0 ?new W():null so 0<=0 is true so first condition check then ++ get executed means c++ execute means the value of c change from 0 to 1 means the value of c is 1 and then new W() option get executed so your default constructor executed and print the output c=1 and in main method we have the one more statement W w2=C(w1) this statement get executed means C method

get executed in which parameter is present means static W c(W w) function get executed in this function contain the statement return w.c++ ==1 ? new W(): null here w.c++==1 first execute but here == having higher priority than ++ so w.c==1 get compare first the value of c was 1 before increment so this statement get satisfy after comparison w.c++ get execute so value c is incremented by 1 means the value of c was 1 before increment and after increment c is 2 and then new W() statement get executed means your default constructor get executed and print the value c it is 2 so our output is C=1 and C=2 and so on .

Question3: What will be the output of given code?

```
class Output
{
    final static short i = 2;
    public static int j = 0;
    public static void main (String [] args)
    {
        for (int k = 0; k < 3; k++)
        {
            switch (k)
            {
                case i: System.out.print(" 0 ");
                case i-1: System.out.print(" 1 ");
                case i-2: System.out.print(" 2 ");
            }
        }
    }
}
```

Options

a) 2 1 0 1 0 0

b) 0 1 2 1 2 2

c) 2 1 2 0 1 2

d) 0 1 2

Ans: option C (2 1 2 0 1 2)

Explanation

If we think about the above code we have the three variable declared in program i , j , k the initial value of i=2 and j=0 and k=0 we use the k in for loop means our statement is

```
for(k=0; k<3; k++){
switch(k)
{ case i: S.o.print("0");
case i-1: S.o.print("1");
case i-2: S.o.print("2");
}
}
```

Here initially k=0; so k<3 get satisfy and we pass k in switch switch(k) means switch(0) so we have the case i means case 2 not match case i-1 means case 2-1 means case 1 one match we have the third statement case 2-1 means 2-2 means case 0 get satisfy and we get the result 2 first and when loop get executed second time then k++ happen then value of k is 1 so k<3 also satisfy this value of k pass in switch(k) means switch(1) so our case i means case 2 not satisfy case 2-i means 2-1 means case 1 get satisfy so it will print 1 and we not use the break state after case so second case also get executed and print 2 means we have the output 2 1 2 after k++ happen so value of k=2 means k<3 this statement get satisfy so in switch we have the switch(k) means switch(2) case i means case 2 get executed then all three cases executed because we not use the break in case so our output is 0 1 2 so final output 2 1 2 0 1 2

Static block in java

Important points related with static block.

- 1) Static block initialize only once in program
- 2) Static block execute very first in program even static block execute before main function also.
- 3) We cannot use the non static variable in static block.

Example of Static Block

```

public class Test
{
    static{
        System.out.println("I am static block");
    }
    public static void main(String x[])
    {
        System.out.println("I am in main methhod");
    }
}

```

```

C:\Program Files\Java\jdk1.8.0_291\bin>java Test
I am static block
I am in main methhod
C:\Program Files\Java\jdk1.8.0_291\bin>

```

If we think about the above code we have the class name as Test in this class we have the one static block and one main method when we run the program then static block get executed before main function and after that main function get executed means a static block get execute very first before main function also.

What is the purpose of static block?

Static block normally use in program when we want to load the code only once in application at the time of application loading

Example: suppose we want to load the configuration from xml at the time of application loading.

Question1: what will be the output of given code?

```

class B {
    static int i;
    static int j;
    static {
        i = 15;
        j = i - 5;
    }
}

```

```

public static void main(String[] args) {
    int i = 0;
    A a = null;
    for (; i < 3; i++) {
        a = new A();
    }
}

```

```
a.i = B.i;  
B.i += a.add(a.operate(i));  
}  
System.out.println(B.i + " " + B.j + " " + i + " " + a.i);  
}  
}
```

```
class A {
```

```
int i = 0;
```

```
int operate(int i) {  
if (B.j - i == i * i * i) return -i;  
return i * i;  
}
```

```
int add(int i) {  
return this.i + i;  
}  
}
```

Options

- a) 14 10 3 0
- b) 14 10 3 16
- c) 120 10 3 61
- d) Compile Time Error

Answer: Option C (120 10 3 61)

Explanation: I will explain in class room (and I will share its video in app);

Question2: what will be the output of given code?

```
class Increment  
{  
public static void main(String[] args)  
{ Simple s = new Simple();  
Simple r = new Simple();  
s.incr();  
r.incr();
```

```

s.display();
r.display();
}
}
class Simple{
static int a = 5;
void incr()
{ a++;
}
void display()
{ System.out.println("a = " + a);
}
}

```

Options

- a) a = 5
a = 5
- b) a = 6
a = 6
- c) a = 7
a = 7
- d) Compilation Error

Output: Option C (a=7 a=7)

Explanation

In this example we have the two classes Increment and Simple in Simple class we have the static variable name as and its initial value is 5 means a=5 and we have the two function name as incr() and display() in incr() function we increase the value of a by 1 and display function we display the value of variable a .

In Increment class we have the statements

```
Simple s = new Simple ();
```

```
Simple r = new Simple ();
```

When we call s.incr() method then value of variable a incremented by 1 means a=6 and we have the one more statement r.incr() then again value of a is incremented by 1 means a=7 and variable use the last existing value

Note: static variable share its common copy between multiple objects of same class means s and r objects or references use the last of value a i.e 7 means when we call the s.display() and r.display() then we get the output of a=7 with s object and a=7 with r object so our output is a=7 and a=7

Question3: what will be the output of given code

```
public class Cricket {  
    static boolean ball;  
    public static void main(String[] args) {  
        short bat = 42;  
        if (bat < 50 & !ball)  
            bat++;  
        if (bat > 50)  
            ;  
        else if (bat > 40) {  
            bat += 7;  
            bat++;  
        }  
        else  
            --bat;  
        System.out.println(bat);  
    }  
}
```

Options

- a) 41
- b) 42
- c) 51
- d) Compilation Error or Runtime Error

Ans: Option C (51)

Explanation: I will solve this problem in class room

Question4: what will be the output of given code?

```

public class GuessCondition
{
    static int a = 40;
    public static void main(String args[])
    {
        System.out.print(a + " ");
        add();
        System.out.print(a);
    }
    private static void add()
    {
        a = a + 40;
    }
}

```

Options

-
- a) 40 40
 - b) 0 0
 - c) 0 40
 - d) 40 80

Correct Ans: Option C (40 80)

Explanation:

As per our code we have the class name as GuessCondition in this class we have the two functions name as public static void main(String args[]) and add() and we declare the one static variable name as a=400

In main we have the statement System.out.print(a+" "); this statement print the value of a i.e 40 and when we call the statement name as add() means we call the add function so your control jump from main method to add function definition and add function we have the statement a=a+40 means we increase the value a by 40 so the current value of a is 80 we print the value of variable a after calling point of add() so it will print the value of a 80 so we have the output is 40 80

Question5: what will be the output of given code ?

```

public class Student {
    int rollno;
    String name;
}

```

```
static String college = "RITA";
static void chage() {
    college = "SRIT";
}
Student(int r, String n) {
    rollno = r;
    name = n;
}
void display() {
    System.out.println(rollno + " " + name + " " + college);
}
public static void main(String arts[]) {
    Student.chage();
    Student s1 = new Student(516, "Kiran");
    Student s2 = new Student(560, "Vishwanath");
    Student s3 = new Student(517, "Kranthi");
    s1.display();
    s3.display();
    s2.display();
}
}
```

Options

- a) 516 Kiran SRIT
516 Kiran SRIT
516 Kiran SRIT
- b) 516 Kiran SRIT
517 Kranthi SRIT
560 Vishwanath SRIT
- c) 516 Kiran RITA
516 Kiran RITA
516 Kiran RITA
- d) 516 Kiran RITA
517 Kranthi RITA
560 Vishwanath RITA

Correct Answer : Option B

516 Kiran SRIT

517 Kranthi SRIT

560 Vishwanath SRIT

Explanation:

As per our code we have the class name as Student with three fields rollno ,name and college we initialize the with default value name as RITA and in this class we have the one static function name as change() and in this function we change the value of college=SRIT and if we think about the main method of class in main method we call the function change using classname because it is a static means we call change using Student.change() when we call this statement then college name change from RITA to SRIT

In main method we have the following statement

Student s1 = new Student(516, "Kiran"); using this statement we initialize the object s1 with 516 as rollno and name as kiran

Student s2 = new Student(560, "Vishwanath"); using this statement we initialize the object s2 with 560 as rollno and vishwanath as name

Student s3 = new Student(517, "Kranthi"); using this statement we initialize the object s3 using 517 as rollno and Kranti as name

Note: college is a static and we change its value from RITA to SRIT so the value of college is SRIT and it is static variable so it is common in all objects of class So when we call s1.display(), s3.display(); , s2.display(); we get the value of every object and college name common with all objects.

Question 6: what will be the output of given code?

```
public class Karbon {
    static int i = 5;
    public static void main(String[] args) {
        Karbon test = null;
        System.out.print("i = " + test.i + ", ");
        System.out.print("i = " + Karbon.i);
    }
}
```

Options

- a) i = 0, i = 5
- b) i = 5, i = 5
- c) Compilation Error
- d) NullPointerException

Explanation:

If we think about the above code we have the class name as Karbon and in this class contain the variable name as i and it is static variable and we initialize the value of variable i=5 and i allocate its memory before creating object of Karbon class means we can use the i without object of Karbon class so if we think about the main method then we have the following statement in main method
 Karbon test =null; here we create the reference of Karbon class we not create its object so we can access the static variable by using reference of class also
 So we have the statement in main System.out.print("I= "+test.i+","); this statement print the I=5 and we have the one more statement in main function name as System.out.print("I =" +Karbon.i); this statement print the value of i=5 because we can use the static variable by using classname also so we have the output like as I=5 I=5

Question7: what will be the output of given code?

```
public class CanYouDolt
{
    static boolean bool;
    static int[] iary = new int[1];
    static char chr;
    static boolean[] barr = new boolean[1];
    public static void main(String args[]) {
        boolean b = false;
        if (bool) {
            b = (chr == iary[chr]);
        } else {
            b = (barr[chr] == bool);
        }
        System.out.println(b + " " + barr[chr]);
    }
}
```

}

Options

- a) false false
- b) true true
- c) false true
- d) true false

Ans: option a (false ,false)

Explanation: I will explain code in class room

Question 8: what will be the output of given code?

```
public class Sketch {  
    static int ad = 100;  
    static int bc = 200;  
    static {  
        ad += 1;  
        bc += 1;  
    }  
    public static void main(String args[]) {  
        ad += 5;  
        bc += 10;  
        System.out.println(ad + bc);  
    }  
    static {  
        ad += 100;  
        bc += 200;  
    }  
}
```

Options

- a) 317
- b) 615
- c) 617
- d) 315

Ans: option b (617)

Explanation: I will explain in classroom.

Question 9: what will be the output of given code?

```

public class KeepItClean {
    public static void main(String[] args) {
        Valuable.status();
        try(Valuable v = new Valuable(); Valuable v2 = new Valuable())
        {
            Valuable.status();
            v.open();
            Valuable.status();
            v2.open();
            Valuable.status();
        }
        Valuable.status();
    }
}

class Valuable implements AutoCloseable {
    static int numberOfValuables = 5;
    Valuable() {
        open();
    }
    public void open() {
        numberOfValuables--;
        System.out.print("O" + numberOfValuables + " ");
    }
    public void close() {
        numberOfValuables++;
        System.out.print("C" + numberOfValuables + " ");
    }
    static void status() {
        System.out.print("S" + numberOfValuables + " ");
    }
}

```

Options

- a) S5 S5 O4 S4 O3 S3 S3
- b) S5 S5 O4 S4 O3 S3 C4 c5 S5

- c) S5 O4 O3 S3 O2 S2 O1 S1 S1
- d) S5 O4 O3 S3 O2 S2 O1 S1 C2 C3 S3

Ans: Option D (S5 O4 O3 S3 O2 S2 O1 S1 C2 C3 S3)

MCQ Question on local variable

Question1: what will be the output of given code?

```
public class ScopeOfVariable {  
    public static void main(String[] args) {  
        int x = 10;  
        int y = 20;  
        {  
            System.out.print(x + " , " + y);  
        }  
        {  
            x = 15;  
            System.out.print(" - " + x + " , " + y);  
        }  
        System.out.print(" - " + x + " , " + y);  
    }  
}
```

Options

- a) 10, 20 - 15, 20 - 15, 20
- b) 10, 20 - 15, 20 - 10, 20
- c) Compilation Error
- d) Runtime Error

Correct Ans: a

Explanation: in above example we have the two variables name as x and y and these are the local variables in main function and we initialize the initial value in x=10 and y=20 we have the statement { System.out.print(x+", "+y) } here we print the values of x and y so we get the answer 10 , 20 then we have one more block { x=15 System.out.print("-"+x+", "+y) } here we initialize the value to x=15 so variable always use the last existing value so x use the 15 value so when we print

the value using `System.out.print("-"+x+", "+y)` so x print the -15 and y print the 20 previous value because we not initialize the value in x and y and we have the last statement `System.out.print("-"+x+", "+y)` here x print -15 and y print the 20 so we have the final output is 10 20 -15 20 -15 20.

Question2: what will be the output of given code?

```
public class UniversalMusic {
    public static void main(String[] args) {
        /*A*/ int pop = 'P';
        /*B*/ char reggae = 'R';
        {
            /*C*/ System.out.println(pop + " " + reggae + " " + hiphop);
            /*D*/ float hiphop = 31.3f;
            /*E*/ System.out.println(pop + " " + reggae + " " + hiphop);
            /*F*/ double jazz = hiphop;
        }
        /*G*/ System.out.println(pop + " " + reggae + " " + hiphop);
        /*H*/ System.out.println(jazz);
        {
            /*I*/ short hiphop = 850062;
            /*J*/ System.out.println(pop + " " + reggae + " " + hiphop);
        }
        /*K*/ System.out.println(pop + " " + reggae + " " + hiphop);
    }
}
```

Options

-
- a) A, C, G, H, K
 - b) C, G, H, K
 - c) C, G, H, I, K
 - d) Compilation error

Correct Answer: d

Explanation: above program generate the compilation error to us because variable hiphop we declare within first { } and closing block and we try to use this variable outside of block so local variable cannot access outside of his block so compile generate the compilation error to us cannot find the symbol.

Question3 (what will be the output of given code)

```
public class Industries {  
    public static void main(String[] args) {  
        String name = "TEXMO";  
        {  
            System.out.println(name + " Industries");  
            name = "texmo";  
        }  
        System.out.println(name + " Industries");  
    }  
}
```

Options

- a) TEXMO Industries
texmo Industries
- b) TEXMO Industries
TEXMO Industries
- c) Compilation Error
- d) Runtime Error

Correct Answer: a

Explanation : if we think about code

```
String name = "TEXMO";  
{  
    System.out.println(name + " Industries");  
    name = "texmo";  
}
```

We have the code here name = "TEXMO" here name use the TEXMO value and we print the values using System.out.println(name+" Industries") so we get output TEXMO industries and we again initialize the variable name using small texmo and we print the variable outside of block using System.out.println(name +"industries"); so we have the output texmo industries.

Question4: what will be the output of above code

```
public class LifeTime {  
    public static void main(String[] args) {  
        if (true) {  
            int x = 10;  
            System.out.println("Value of X = " + x);  
            x++;  
        }  
    }  
}
```

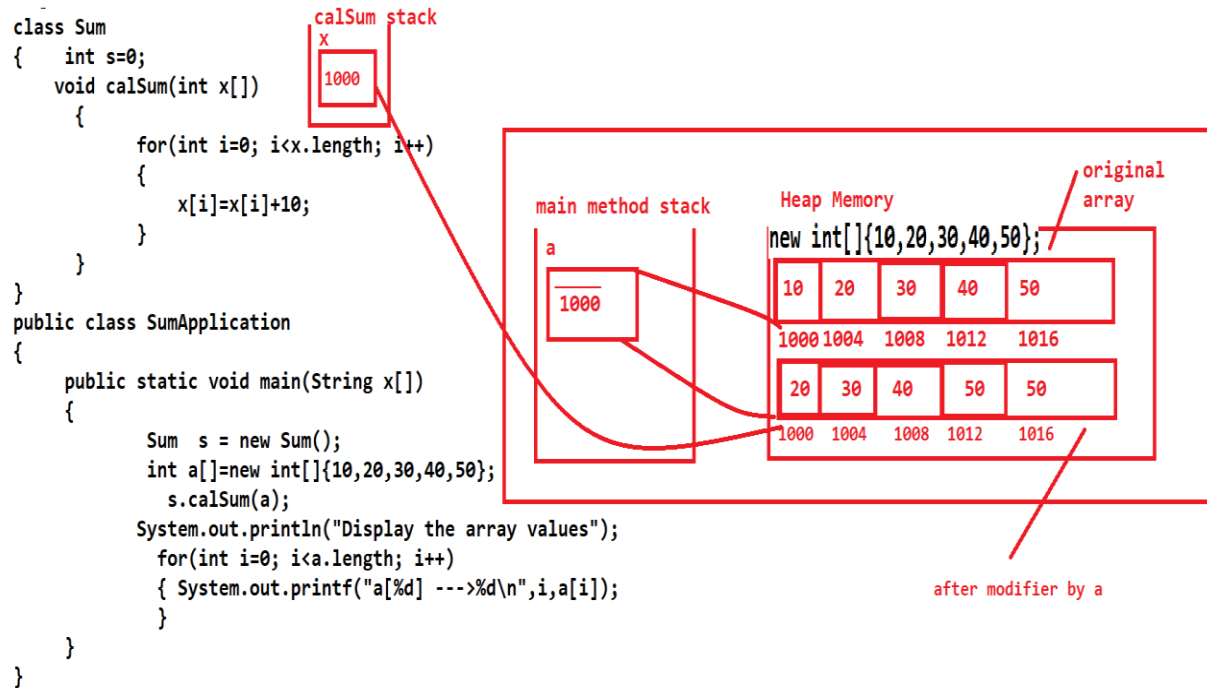
Options

- a) Value of X = 10
- b) Value of X = 0
- c) Compilation Error
- d) Runtime Error

Correct Answer: a

Explanation: if we think about the above code we have the statement name as if (true) this statement indicate we have the if statement is true so if get satisfied and we initialize the value x=10 so the value of x is 10 and when we print the value using System.out.println ("Value of x = "+x) so it will print the value of x is 10.

MCQ Question on array as parameter using java



Output

```

D:\evenigadvjava>javac SumApplication.java

D:\evenigadvjava>java SumApplication
Display the array values
a[0] --->20
a[1] --->30
a[2] --->40
a[3] --->50
a[4] --->60

D:\evenigadvjava>

```

Explanation of above code

If we think about the above code we have the two classes name as Sum and SumApplication in Sum class we have the function or method name as calSum(int

x[]) which contain the array as parameter and in main method we have array name as `int a[] = new int[]{10,20,30,40,50}` as per this statement array allocate its memory in heap section and its reference present in main function and store the base address of array present in heap section of memory when we call the method name as `calSum(a)` then base address of array a pass to `calSum(int x[])` function then x array store the base address of a means x points to the array a stored in memory and in `calSum()` function we have the statement

```
for (int i=0; i<x.length; i++) {
    x[i] =x[i]+10;
}
```

This statement modify the value of array x means indirectly we modify the value of array a and increment every value by 10 and stored in array and in main function when we print the value of array a then we get the answer shown in above output.

WAP to Create the class Name as Company with method name as `addNewEmployee()` with parameter name , id ,sal ,qualification and one more method name as `showDetail()` and display the employee details using this method.

Source code

```
class Company
{   private String name;
    private int id;
    private int sal;
    private String qual;
    public void addNewEmployee(String name,int id,int sal, String qual)
    {   this.name=name;
        this.id=id;
        this.sal=sal;
        this.qual=qual;
    }
    public void show ()
    {   System.out.println (name+"\t"+id+"\t"+sal+"\t"+qual);
    }
}
```

```

public class CompanyApplication
{
    public static void main (String x [])
    {
        Company c = new Company ();
        c.addNewEmployee ("Ram", 1, 10000,"BE");
        c.show ();
    }
}

```

Code Explanation

If we think about the above code we have the two classes name as Company and CompanyApplication in CompanyApplication we create the object of Company class and in using this object we call the method name as addNewEmployee () in this method we pass the four parameter and store the data in object and we call the show () method for display the record of employee.

The major limitation of above code is if we have the number of parameter in addNewEmployee() like as name ,id,sal,qual,address,preexp,expsal,pan,adhar etc then it is verify difficult to pass parameter and remember its sequence at function calling point so if we want to avoid this problem java provide the POJO class concept

MCQ Question on POJO class

```

class Player
{
    private int id;
    private String name;
    public void setId(int id)
    {
        this.id=id;
    }
    public int getId()
    {
        return id;
    }
    public void setName(String name)
    {
        this.name=name;
    }
    public String getName()
    {
        return name;
    }
}

```

```

    }
}
class Team
{
    public void addPlayer(Player player)
    {
        player.setId(100);
        player.setName("Dhoni");
    }
}
public class PlayerApplication
{ public static void main(String x[])
    {   Player p = new Player();
        p.setId(1);
        p.setName("ABC");
        Team t =new Team();
        t.addPlayer(p);
        System.out.println(p.getId()+"\t"+p.getName());
    }
}

```

Output

```

C:\Program Files\Java\jdk1.8.0_291\bin>java PlayerApplication
100      Dhoni

C:\Program Files\Java\jdk1.8.0_291\bin>

```

Explanation

If we think about the above code then we have three class's one is Player which is POJO class and we use it for storing the information about the Player and in this class we have the two field name and id. We have the one more class name as Team and in Team class we have the method name as addNewPlayer(Team t) in this method we pass reference of Player class in addNewPlayer() and In

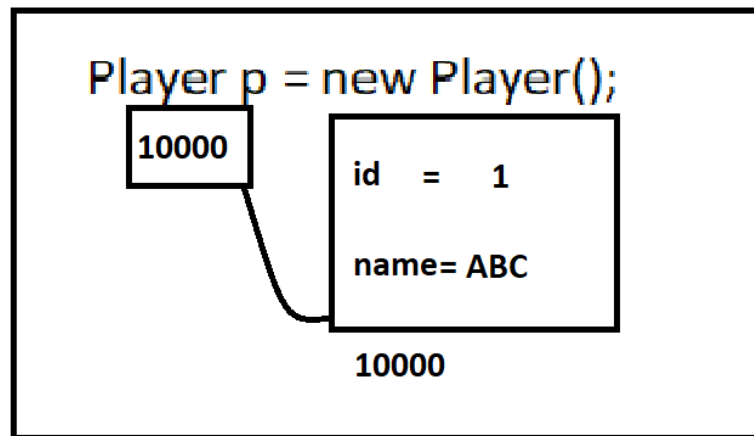
PlayerApplication class contain main method so first now we will discuss about the main method present in PlayerApplication class.

In main method first we have the following statement

```
Player p = new Player();
p.setId(1);
p.setName("ABC");
```

as per above code we create the object of player class and store the id and name in Player class object shown in following diagram.

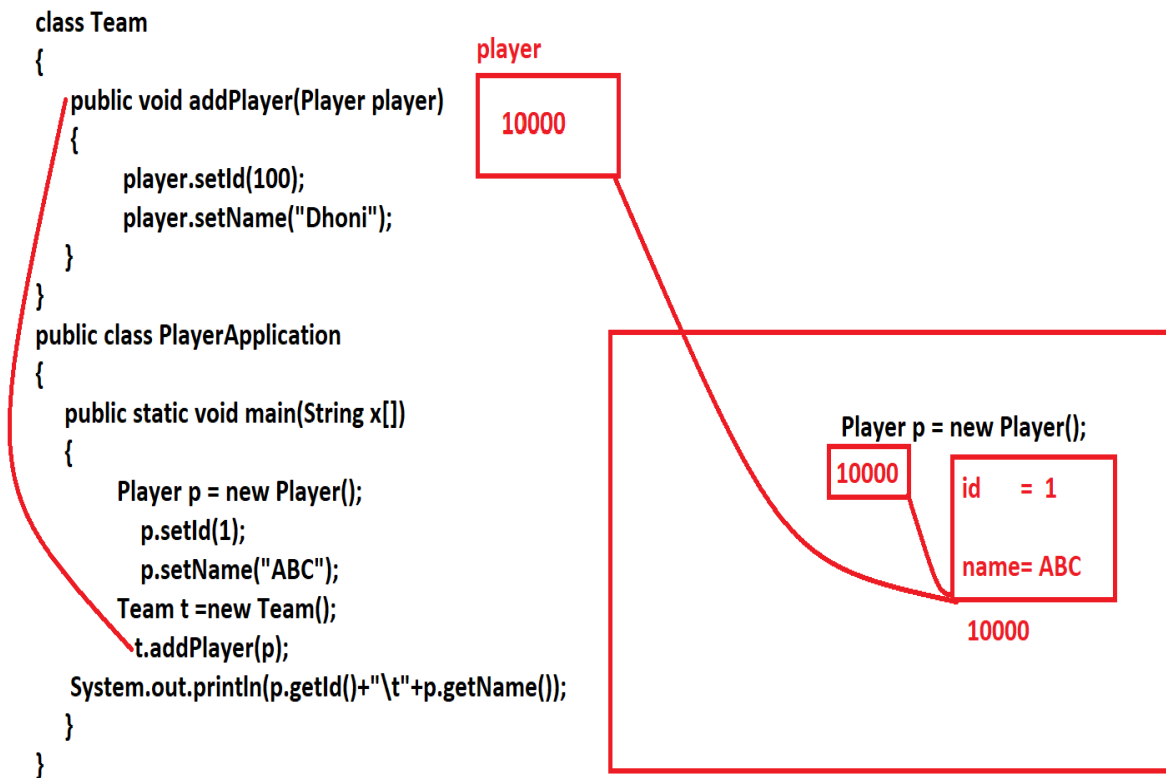
```
Player p = new Player();
p.setId(1);
p.setName("ABC");
```



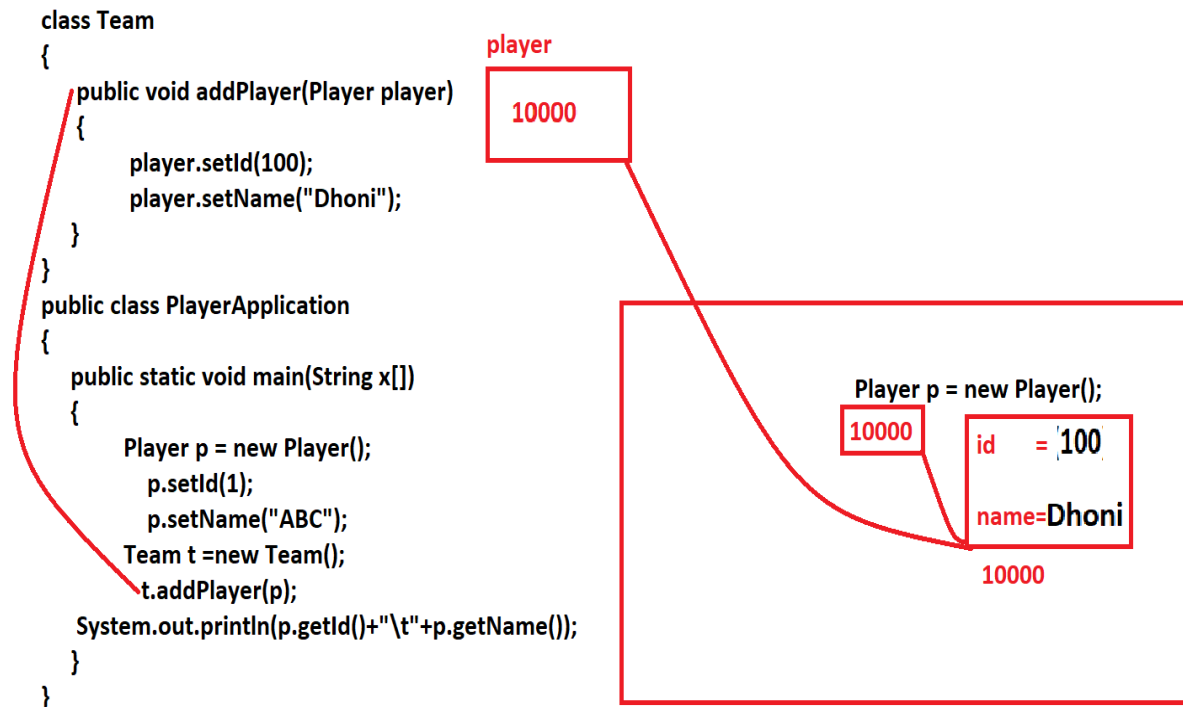
Here we create the object of Player class whose address is 10000 as per our example and we store this address in reference name as p after that we have the statement

```
Team t = new Team();
t.addPlayer(p);
```

as per above code we create the object of Team class Team t = new Team and we call the method name as t.addPlayer(p) means we pass the reference of Player class to Team class method addPlayer() shown in following diagram.



When we pass the reference of Player to `addPlayer()` method of Team class means as per our example reference `p` present in `main` method and reference Player present in `addPlayer()` method points to same the memory i.e 10000 addresses as per our example so in `addPlayer()` method we have the two statements `player.setId(100)` and `player.setName("Dhoni")` means we modify the content on 10000 address space from `addPlayer()` method means so 100 and Dhoni get override on 10000 address space shown in following diagram.



After `t.addPlayer (p)` we have the statement given below
`System.out.println (p.getId () +"\t"+p.getName ());` this statement print the 100 and dhoni so our program print the output 100 and dhoni.

Array of Objects concept in java

Array of objects is used for store the more than one object data in single name reference called as array of objects.

How to create the array of objects in java

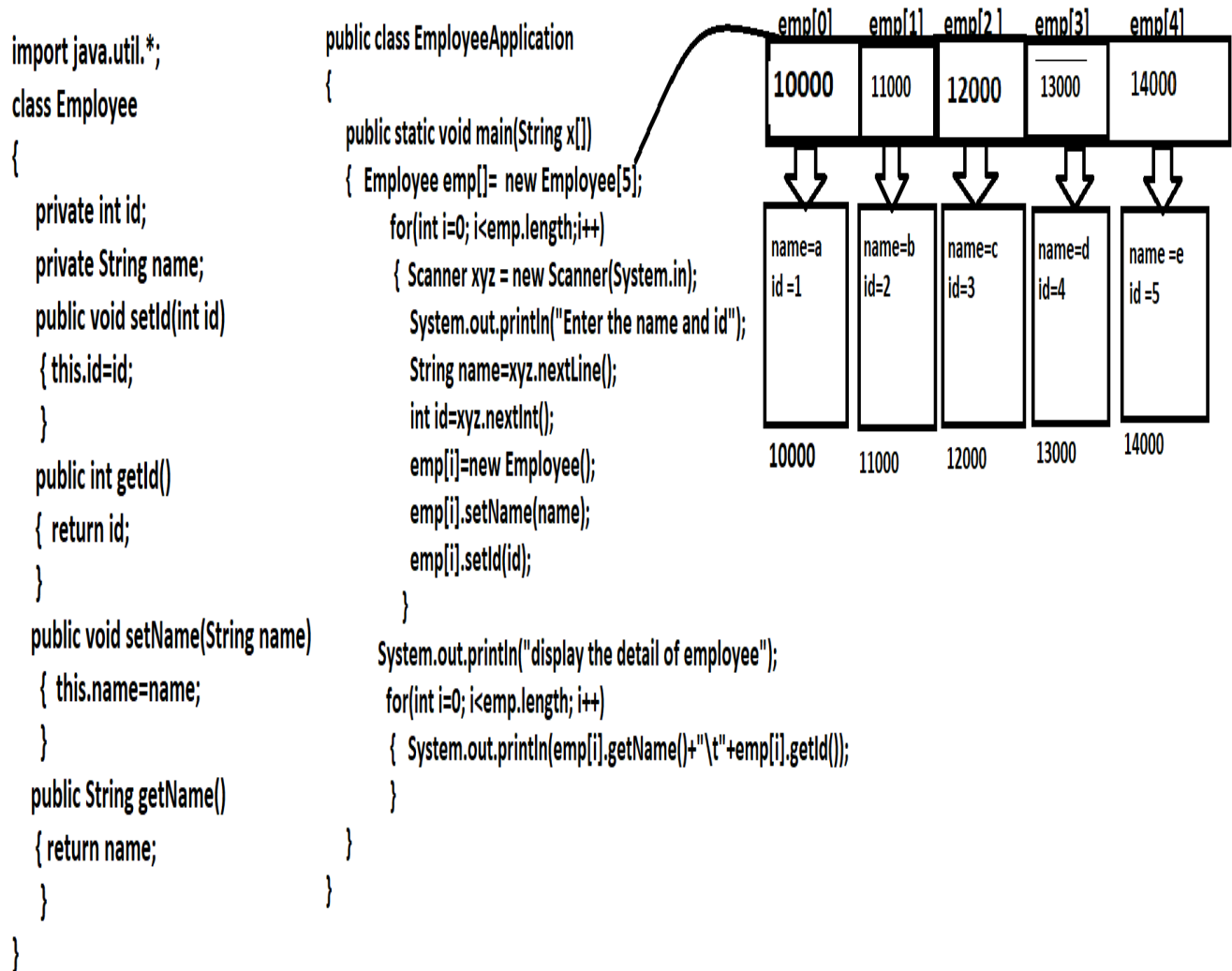
```

classname ref[]=new classname[size] ; //this array of reference
for (int i=0; i<ref.length; i++)
{
    ref[i] = new classname (); // array of objects
}

```

When we create the array of objects we have to create the array of reference because if we have multiple objects then we have to create the number of references for store that objects.

Example: suppose consider we have the Employee class with field id and name and we want to create the five employee objects store data in it and display its record.



Output

```
C:\Program Files\Java\jdk1.8.0_291\bin>java EmployeeApplication
Enter the name and id
a
1
Enter the name and id
b
2
Enter the name and id
c
3
Enter the name and id
d
4
Enter the name and id
e
5
display the detail of employee
a      1
b      2
c      3
d      4
e      5
```

Source code of above diagram

```
import java.util.*;
class Employee
{
    private int id;
    private String name;
    public void setId(int id)
    { this.id=id;
    }
    public int getId()
    { return id;
    }
    public void setName(String name)
    { this.name=name;
    }
}
```

```

    }
    public String getName()
    { return name;
    }
}
public class EmployeeApplication
{
    public static void main(String x[])
    {
        Employee emp[]= new Employee[5];
        for(int i=0; i<emp.length;i++)
        { Scanner xyz = new Scanner(System.in);
          System.out.println("Enter the name and id");
          String name=xyz.nextLine();
          int id=xyz.nextInt();
          emp[i]=new Employee();
          emp[i].setName(name);
          emp[i].setId(id);
        }
        System.out.println("display the detail of employee");
        for(int i=0; i<emp.length; i++)
        { System.out.println(emp[i].getName()+"\t"+emp[i].getId());
        }
    }
}

```

Description of above code

If we think about the above code we have the two classes name as Employee it is our POJO class and EmployeeApplication class which contain main method in main method class we have the statement

Employee emp[]=new Employee[5] this statement indicate the array of reference of Employee class means we have the 5 references because we want to store the five object address in it and we have the for loop given below

```

for(int i=0; i<emp.length;i++)
{ Scanner xyz = new Scanner(System.in);
  System.out.println("Enter the name and id");

```

```

        String name=xyz.nextLine();
        int id=xyz.nextInt();
        emp[i]=new Employee();
        emp[i].setName(name);
        emp[i].setId(id);
    }

```

For(int i=0; i<emp.length; i++): this statement travel your for loop five times after that we have the statement name as Scanner xyz = new Scanner(System.in) this statement indicate we have the scanner class for input after that we have the statement String name=xyz.nextLine() for accept the input of type string int id=xyz.nextInt() for accept the input of type integer emp[i]=new Employee(): this statement indicate we create the new object of Employee every time and store its address on array specified index after that we have the statement emp[i].setName(name) and emp[i].setId(id) this statement indicate we store the data in object using its index

And we have the one more for loop data fetching from array of objects.

```

    for(int i=0; i<emp.length; i++)
    {   System.out.println(emp[i].getName()+"\t"+emp[i].getId());
    }

```
