Methods

Find Output

```
class Main {
    public static void main(String args[]) {
            System.out.println(fun());
    }

    int fun()
    {
      return 20;
    }
}
```

Output

(A) 20
(B) compiler error
(C) 0
(D) garbage balue

Answer: (B)

Explanation: main() is a static method and fun() is a non-static method in class Main.

Find Output

```
public class Test {
public static void main(String[] args) {
    String x = null;
    giveMeAString(x);
    System.out.println(x);
  }
  static void giveMeAString(String x)
  {
    x = "raju";
  }
}
```

Output

(A) raju
(B) null
(C) Compiler Error
(D) Exception

Answer: (B)

Explanation: Parameters in Java is passed by value. So the changes made to x do not reflect in main().

Find Output

```
class Test {
public static void swap(Integer i, Integer j) {
    Integer temp = new Integer(i);
    i = j;
    j = temp;
  }
  public static void main(String[] args) {
    Integer i = new Integer(10);
    Integer j = new Integer(20);
    swap(i, j);
    System.out.println("i = " + i + ", j = " + j);
  }
}
```

Output

(A) i = 10, j = 20
(B) i = 20, j = 10
(C) i = 10, j = 10
(D) i = 20, j = 20

Answer: (A)

Explanation: Parameters are passed by value in Java

Find Output

```
class intWrap {
  int x;
}
public class Main {
   public static void main(String[] args) {
     intWrap i = new intWrap();
     i.x = 10;
     intWrap j = new intWrap();
     j.x = 20;
     swap(i, j);
     System.out.println("i.x = " + i.x + ", j.x = " + j.x);
   }
   public static void swap(intWrap i, intWrap j) {
     int temp = i.x;
     i.x = j.x;
     j.x = temp;
   }
}
```

Output

(A) i.x = 20, j.x = 10
(B) i.x = 10, j.x = 20
(C) i.x = 10, j.x = 10
(D) i.x = 20, j.x = 20

Answer: (A)

Explanation: Objects are never passed at all. Only references are passed. The values of variables are always primitives or references, never objects

Find Output

```
class Main {
    public static void main(String args[]) {
            System.out.println(fun());
    }
    static int fun(int x = 0)
    {
      return x;
    }
}
```

Output

(A) 0
(B) Garbage Value
(C) Compiler Error
(D) Runtime Error

Answer: (C)

Explanation: Java doesn’t support default arguments. In Java, we must write two different functions.

Find Output

```
public class Test {
public static void main(String[] args) {
        String str = "java";
        str.toUpperCase();
        str += "concepts";
        String string = str.substring(2,8);
        string = string + str.charAt(4);;
        System.out.println(string);
}
}
```

Output

- Options:

a) vaconcc

b) VAconcc

c) javaconcepts

d) Vaconco

Ans: a

Find Output

What is the process of defining more than one method in a class differentiated by method signature?

a) method overriding

b) method overloading

c) method doubling

d) None of the mentioned

Output

- Ans: b

Find Output

```
public class Test {
public static void sum(int ...x)
   {
        int sum=0;
        for(int n:x)
        {
              sum=sum+n;
        }
        System.out.print(sum+” “);
   }
public static void main(String[] args) {
       sum(10,20,30);
       sum(10,20,30,40);
}
}
```

Output

- Options:

a) 60 100

b) 160

c) Compilation error

d) 10 10

Ans) a

Find Output

```
class Area
  {
    int width,length,volume,height;
    Area()
    {
      width=5;
      length=6;
    }
    void volume()
    {
       volume = width*length*height;
    }
  }
public class Test {
public static void main(String[] args) {
    Area obj = new Area();
    obj.volume();
    System.out.println(obj.volume);
}
}
```

Output

a) 0
b) 1
c) 30
d) Compilation error

Ans: a

Find Output

```
class Equality
    {
        int x;
        int y;
        boolean isequal()
        {
            return(x == y);
        }
    }
public class Test {
public static void main(String[] args) {
        Equality obj = new Equality();
        obj.x = 5;
        obj.y = 5;
        System.out.println(obj.isequal());
}
}
```

Output

a) false
b) true
c) 0
d) 1

- Answer: b

Find Output

Output

Find Output

Output