Predict the output for the following code.

```
1. public class Main{
public static void main(String[] list){
System.out.println (“Hello main”);
}
}
```

a. Error
b. Output : Hello main

Ans: Hello main

```
2.
public class Main{
public void main(String[] list){
System.out.println (“Hello main”);
}
}
```

a. Error
b. Output : Hello main

Ans: RunTimeError:- Main method is not static in Main Class (static is missing there)

```
3. public class Main{
public static main(String[] list){
System.out.println (“Hello main”);
}
}
```

a. Error
b. Output : Hello main

Ans: RunTimeError:- Main method is not found in Main Class (return type void is missing there)

```
4. public class Main{
static void main(String[] list){
System.out.println (“Hello main”);
}
}
```

a. Error
b. Output : Hello main

Ans: RunTimeError:- Main method is not found in Main Class (Access specifier public is missing there)

```
5. public class Main{
        int var = 110;
        public static void main(String[] list){
        System.out.println (“Hello main”);
        System.out.println (“var : ” + var);
        }
}
```

a. Errorb. Output : Hello main
var : 110

Ans: Error: non-static variable cannot be referenced from static context

Explanation: Var is declared in class Main and it is instance variable i.e non static. main() method is static so we cann't access non- static in static contest

6.

```
public class Main{
int var = 110;
public void main(String[] list){
int var = 220;
System.out.println ("Hello main");
System.out.println ("var : " + var);
}
}
```

a. Error
b. Output : Hello main
var : 220

Ans: b

7.

```
public class Main{
int var = 110;
static int staticvar = 1100;
public void main(String[] list){
System.out.println ("Hello main");
System.out.println ("var : " + staticvar);
}
}
```

a. Error
b. Output : Hello main
var : 1100

Ans:b

8.

```
public class Main{
int var = 110;
static int staticvar = 1100;
public void main(String[] list){
Main obj = new Main();
System.out.println ("Hello main");
System.out.println ("var : " + staticvar);
System.out.println ("var : " + obj.var);
}
}
```

a. Error
b. Output : Hello main
var : 1100
var : 110

Ans:b

Static content can be referenced by using object and directly in static method but to access non-static i.e instance variable in static method then we have to create object of that class and then access it by using (.) operator

9.public class Main{
int var = 110;
static int var = 1100;
public void main(String[] list){
int var = 220;
System.out.println (“Hello main”);
System.out.println (“var : ” + var);
}
}
a. Error
b. Output : Hello main
var : 110
c. Output : Hello main
var : 220
d. Output : Hello main
var : 1100

Ans: Error:

We cannot give same name to the variables in the same block it gives error that already defined

10. public class Main{
int var1 = 121;
static int staticvar1 = 1212;
public void demo(){
System.out.println (“Demo function”);
System.out.println (var1);
System.out.println (staticvar1);
}
public void main(String[] list){
Main obj = new Main();
System.out.println (“Hello main”);
obj.demo();
}
}
a. Error
b. Output : Hello main
Demo function
121
1212
Ans: b

11. public class Main{
int var1 = 121;
static int staticvar1 = 1212;
public void demo(int var1){
System.out.println (“Demo function”);System.out.println (var1);
System.out.println (staticvar1);
}
public void main(String[] list){

int var1 = 121212;
Main obj = new Main();
System.out.println (“Hello main”);
obj.demo(var1);
}
}
a. Error
b. Output : Hello main
Demo function
121212
1212
c. Output : Hello main
Demo function
121
1212

Ans: b

12. public class Main{
int var1 = 121;
static int staticvar1 = 1212;
public void main(String[] list){
int var1 = 121212;
System.out.println (“Hello main ” + var1);
}
}
a. Error
b. Output : Hello main 121212
c. Output : Hello main 121

Ans: b

13. public class Main{
int var1 = 121;
static int staticvar1 = 1212;
public void demo(){
System.out.println (“Demo function”);
System.out.println (var1++);
System.out.println (staticvar1++);
}
public void main(String[] list){
Main obj = new Main();System.out.println (“Hello main”);
obj.demo();
}
}
a. Error
b. Output : Hello main
Demo function
121
1212
c. None of these

Ans: b

14. public class Main{
int var1 = 121;
static int staticvar1 = 1212;
public void demo(){
System.out.println (“Demo function”);
System.out.println (++var1);
System.out.println (++staticvar1);
}
public void main(String[] list){
Main obj = new Main();
System.out.println (“Hello main”);
obj.demo();
}
}
a. Error
b. Output : Hello main
Demo function
121
1212
c. None of these

Ans: c
because the ++var1 will return incremented value i.e. 122

15. public class Main{
int var1 = 121;
static int staticvar1 = 1212;
public void demo(){
System.out.println (“Demo function”);
var1++;
staticvar1++;
System.out.println (var1);
System.out.println (staticvar1);
}public void main(String[] list){
Main obj = new Main();
System.out.println (“Hello main”);
obj.demo();
Main obj2 = new Main();
obj2.demo();
}
}
a. Error
b. Output : Hello main
Demo function
122
1213
Demo function
122
1214
c. None of these

d. Output : Hello main
Demo function
122
1213
Demo function
123
1214
Ans:d

16. public class Main{
public static void staticdemo(){
System.out.println(“Static method”);
}
public void demo(){
System.out.println (“Demo function”);
}
public void main(String[] list){
Main obj = new Main();
System.out.println (“Hello main”);
staticdemo();
obj.demo();
}
}
a. Error
b. Output: Hello main
Static method
Demo function
Ans: b

17. public class Main{
static staticblk(){
System.out.println(“Staticblk”);
}
public static void staticdemo(){
System.out.println(“Static method”);
}
public void demo(){
System.out.println (“Demo function”);
}
public void main(String[] list){
Main obj = new Main();
System.out.println (“Hello main”);
staticdemo();
obj.demo();
}
}
a. Error
b. Output: Staticblk
Hello main
Static method
Demo function

error: invalid method declaration; return type required
static staticblk(){

18. public class Main{
static{
System.out.println(“Staticblk”);
}
public static void staticdemo(){
System.out.println(“Static method”);
}
public void demo(){
System.out.println (“Demo function”);
}
public void main(String[] list){
Main obj = new Main();
System.out.println (“Hello main”);
staticdemo();
obj.demo();
}
}
a. Error
b. Output: StaticblkHello main
Static method
Demo function
Ans:b

19. public class Main{
public static void main(int var){
System.out.println(“Another main”);
}
public void main(String[] list){
Main obj = new Main();
System.out.println (“Hello main”);
obj.main();
}
}
a. Error
b. Output: Hello main
Another main
c. None of these

Ans:Error: no method found for main() because it requires one int parameter

20. public class Main{
public static void main(int var){
System.out.println(“Another main”);
}
public void main(String[] list){
System.out.println (“Hello main”);
}
}

a. Error
b. Output: Hello main
c. Output: Hello main
Another main
d. None of these

Ans:b

21. public class Main{
static {
System.out.println(“Staticblk”);
}
}
a. Error
b. Output: Staticblk
c. None of these
Ans:a

22. public class Main{
Main(){
System.out.println(“Constructor”);
}
public void main(String[] list){
System.out.println (“Hello main”);
Main obj = new Main();
}
}
a. Error
b. Output: Hello main
c. Output: Hello main
Constructor
d. None of these

Ans:c

23. public class Main{
Main(){
System.out.println(“Constructor”);
}
void main(){
System.out.println (“main”);
}
public void main(String[] list){
System.out.println (“Hello main”);
Main obj = new Main();
obj.main();
}
}
a. Error
b. Output: Hello main
c. Output: Hello main
Constructor

main
d. None of these

Ans:c

24. public class Main{
Main obj = new Main();
obj.main();
Main(){
System.out.println(“Constructor”);
}public main(String[] list){
System.out.println (“Hello main”);
}
}
a. Error
b. Output: Hello main
c. Output: Constructor
d. None of these
Ans:d

25. public class Main{
Main obj = new Main();
obj.main();
Main(){
System.out.println(“Constructor”);
}
public void main(){
System.out.println (“main”);
}
public void main(String[] list){
System.out.println (“Hello main”);
}
}
a. Error
b. Output: Hello main
main
c. Output: main
Hello main
Constructor
d. None of these

Ans:a We cannot call any method from non-executable form

26. public class Main{
public static void main(){
System.out.println(“main”);
}
public void main(String[] list){
System.out.println (“Hello main”);
}

}
a. Errorb. Output: Hello main
c. Output: main
d. None of these

Ans: a
main method not found String[] args is not given