Question 1:

Determine the output of the following code when executed with the command
java HelloWorld hello world goodbye
Modify accordingly if compilation fails with an error.

```
1 public static class HelloWorld {
2     public static void main(String[] args){
3         System.out.println(args[1] + args[2]);
4     }
5 }
```

Explanation:
The class declaration on line contains the static modifier, which is not a valid modifier for a top-level class. This causes a compiler error.
How to define the argument input in Eclipse?
Run->Run Configurations->Arguments->Program arguments
File Name – HelloWorld.java

Question 2:

What is the result of the following code? Modify accordingly if compilation fails with an error.

```
1  public class Shape {
2      private String color;
3      public Shape(String color){
4          System.out.print(Shape);
5          this.color = color;
6      }
7      public static void main(String[] args){
8          new Rectangle();
9      }
10 }
11
12 class Rectangle extends Shape {
13     public Rectangle() {
14         System.out.print("Rectangle");
15     }
16 }
```

Explanation:

If a constructor does not call this or super on its first line of code, the compiler inserts the statement super(), which occurs in the Rectangle class just before line System.out.print("Rectangle");. A call to super() in Rectangle invokes a no-argument constructor in Shape, but Shape does not have a no-argument constructor. The compiler error occurs.

File Name – Shape.java

Question 3:

What will be the result of compiling and running the following program? Modify accordingly if compilation fails with an error.

```
1  public class Init {
2      String title;
3      boolean published;
4      static int total;
5      static double maxPrice;
6
7      public static void main(String[] args){
8          Init initMe = new Init();
9          double price;
10         if(true)
11             price = 100.00;
12         System.out.println("|"+initMe.title+"|"+
13             initMe.published+"|"+Init.total+"|"+Init.maxPrice+"|"+price+"|");
14     }
15 }
```

Explanation:

The program will compile. The compiler can figure out that the local variable price will be initialized, since the value of the condition in the if statement is true. The two instance variables and the two static variables are all initialized to the respective default value of their type.

File Name – Init.java

Question 4:

The following program has several errors. Modify the program so that it will compile and run without errors.

```
1  PUBLIC CLASS temperature {
2      PUBLIC void main( string args ) {
3      double fahrenheit = 62.5;
4      */ Convert /*
5      double celsius = f2c(fahrenheit ); System.out.println(fahrenheit + 'F' + " = " + Celsius + 'C');
6      }
7      double f2c(float fahr) {
8      return (fahr - 32) * 5 / 9;
9      }
10 }
```

Solution

1. *PUBLIC is not key word. Correct syntax is public.*
2. *CLASS is not key word. Correct syntax is class.*
3. **/ Convert /* is wrong syntax for comment. Correct syntax is /* Convert */ class.*
4. *Parameter accept by f2c is float, and program tries to pass float.*
5. *Celsius is not declared in the program.*
6. *f2c is non static method that cannot be called in the static context.*

File Name – temperature.java

Question 5:

What will be the result of compiling the following program? Modify accordingly if compilation fails with an error.

```
1 public class MyClass {
2     long var;
3     public void MyClass(long param) { var = param; }
4     public static void main(String[] args){
5         MyClass a,b;
6         a = new MyClass();
7         b = new MyClass(5);
8     }
9 }
```

Explanation:

A compilation error will occur at b = new MyClass(5);, since the class does not have a constructor accepting a single argument of type int. MyClass(long param) declares a method, not a constructor, since it is declared as void. The method happens to have the same name as the class, but that is irrelevant. The class has an implicit default constructor, since the class contains no constructor declarations.

File Name – MyClass.java