The result of trying to compile and execute :the following program is

```
1 public class GetExceptions {
2 public static void main(String[] args) {
3 try{
4    m();
5 }catch (Exception e){
6    System.out.println("Exception Handled in Main");
7 }
8  f();
9 }//end main
10 public static void m() throws Exception {
11 try {
12   System.out.println("Method m");
13   throw new Exception();
14 }catch (Exception e){
15   System.out.println("Handled in method m");
16   throw e;
17 } finally{
18    System.out.println("Finally m");
19 }
20 }//end m
21 public static void f(){
22 try{
23    System.out.println("method f");
24 }catch (Exception e){
25    System.out.println(e.toString());}
26 finally{
27    System.out.print("Finally f");}
28 }//end f
29 }
```

Select the lines that will be printed

- [x] Method m
- [x] Handled in method m
- [x] Finally m
- [x] Exception Handled in Main
- [x] method f
- [x] Finally f

Question 3

The result of trying to compile and execute * :the program is

```
1 class SalaryCalculationException extends Exception{}
2 class Person{
3 public void calculateSalary() throws SalaryCalculationException{
4   throw new SalaryCalculationException();
5 }//end calculateSalary
6 }
7 class Company{
8 public void paySalaries(){
9   Person p = new Person();
10  p.calculateSalary();
11 }//end paySalaries
12 }
```

- Program will not run until we return a boolean instead of void method paySalaries() ○
- Program will not run until we add throws RuntimeException in the signature of method paySalaries() ○
- Program will not run until we add a try-catch block in paySalaries() ◉
- Program runs correctly ○

Question 2

The result of trying to compile and execute

* :the program is

```
1 class point{
2 int x = 0;
3 public void increment(int n){
4   x = x + n;
5 }//end increment
6 }
7 public class MyClass {
8 public static void main(String args[]){
9 point[] plist = new point[5];
10 try{
11   for(int i = 0; i < plist.length; i++)
12      plist[i].increment(i);
13 }
14 catch (ArrayIndexOutOfBoundsException e){
15 System.out.println("Invalid index");
16 }
17 catch (NullPointerException e){
18 System.out.println("Null pointer");}
19 catch (Exception e){
20 System.out.println("Exception");
21 }
22 }//end main
23 }
```

- Invalid index ○
- Null pointer ◉
- Exception ○
- Nothing ○

Question 3

The result of trying to compile and execute the program is: *

```
1  public class test {
2  public static void main(String args[]){
3  try{
4    int a = 1;
5    int c[] = {1};
6    try{
7      c[a] =9;
8    }
9    finally{
10     System.out.print("A");
11   }
12 }
13 catch (Exception e){
14   System.out.println("B");
15 }
16 }//end main
17 }
```

- ○ ArrayIndexOutOfBoundsExceptoin:1 A
- ○ B
- ○ AB
- ○ BA

The result of trying to compile and execute the following program is:

```
1  public class TestExceptions{
2  public static void main(String[] args) throws Exception {
3  try{
4    method2();
5  }
6  catch (RuntimeException e){
7    System.out.println("Runtime Main");
8  }
9  catch (Exception e){
10   System.out.println("Main");
11 }
12 }//end main
13 public static void method1() throws Exception{
14 try{
15   System.out.println("Method 1");
16    throw new Exception();
17 }catch (Exception e){
18   System.out.println("Handled in method 1");
19   throw e;
20 }finally{System.out.println("Finally 1");}
21 }//end method1
22 public static void method2() throws Exception{
23 try{
24   System.out.println("method 2");
25 }catch (Exception e){
26   System.out.println(e.toString());}
27 finally{
28   System.out.print("Finally 2");
29    method1();}
30 }//end method 2
31 }
```

Select the lines that will be printed

- [ ] Method 1
- [ ] method 2
- [ ] Finally 2Method 1
- [ ] Handled in method 1
- [ ] Finally 1
- [ ] Main
- [ ] Finally 2
- [ ] Runtime Main

The result of trying to compile and execute the program is: *

```
class A{
public void function(){
B b = new B();
b.function();
System.out.print("A");
}//end function
}
class B{
public void function(){
C c = new C();
c.function();
System.out.print("B");
}//end function
}
class C{
public void function(){
if(true)
  throw new NullPointerException();
System.out.print("C");
}//end function
}
public class Test{
public static void main(String args[]){
try{
  A a = new A();
  a.function();
}catch(Exception ex){
  System.out.print("error");
}
}//end main
}
```

- ○ CBA
- ○ errorABC
- ○ BAerror
- ○ error