1. What is the output of the following program? (6 points)

```
#include <iostream>
using namespace std;
int main( )
{
        int i, arraySize = 4;
        int array[arraySize];
        int *ip1, *ip2;
        ip1 = array;
        ip2 = new int[arraySize];
        for (i = 0; i < arraySize; i++)
        {
                array[i] = i * 2;
                ip2[i] = i * 3;
        }
        for (i = 0; i < arraySize; i++) cout << ip1[i] << " ";
        cout << endl;
        for (i = 0; i < arraySize; i++) cout << *(ip2 + i) << " ";
        cout << endl;
        return 0;
}
```


2. Fill in the output in the following **Sample Dialogue**. (6 points)

```
#include <iostream>
#include <cctype>
using namespace std;
int main( )
{
        char temp;
        cout << toupper('d') << endl;
        temp = toupper('d');
        cout << temp << endl;
        cout << (char) toupper('d') << endl;
        cout << static_cast<char> (toupper('d')) << endl;
        return 0;
}
```

Sample Dialogue

6	8																

3. Fill in the output in the following **Sample Dialogue**. (4 points)

```
#include <iostream>
using namespace std;
int main()
{
        char one, two, three, four;
        cout << "Enter four characters: ";
        cin.get(one);
        cin.get(two);
        cin.get(three);
        cin.get(four);
        cout << one << two << three << four << "End!";
        return 0;
}
```

Sample Dialogue

E	n	t	e	r		f	o	u	r		c	h	a	r	a	c	t	e	r	s	:		1		3		5		7

4. What is the output of the following program? (6 points)

```
#include <iostream>
using namespace std;
struct Item
{
    int ip;
    int price;
};
int main( )
{
        Item one, two;
        one.ip = 1000;
        one.price = 19;
        cout << one.ip << " $" << one.price << endl;
        two = one;
        two.ip ++;
        two.price *= 2;
        cout << two.ip << " $" << two.price << endl;
        return 0;
}
```


5. Fill in the output in the following **Sample Dialogue**. (3 points)

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
        string first, second("Today");
        cout << "Enter a line of input:" << endl;
        cin >> first;
        if (first == second)
                cout << "Same!" << endl;
        else
                cout << "Different!" << endl;
        return 0;
}
```

Sample Dialogue

E	n	t	e	r		a		l	i	n	e		o	f		i	n	p	u	t	:	
T	**o**	**d**	**a**	**y**		**i**	**s**		**a**		**s**	**h**	**i**	**n**	**y**		**d**	**a**	**y**	**!**		

6. Suppose there is a file **infile.dat**, which of the following statements can open the file correctly? (5 points)

(a)
```
ifstream in;
in.open("infile.dat");
```
(b)
```
ifstream in;
string fileName = "infile.dat";
in.open(fileName);
```
(c)
```
ifstream in;
string fileName = "infile.dat";
inStream.open(fileName.c_str( ));
```
(d)
```
ifstream in;
char fileName[ ] = "infile.dat";
inStream.open(fileName);
```
(e)
```
ifstream in;
char fileName[20] = "infile.dat";
inStream.open(fileName);
```

(a)	(b)	(c)	(d)	(e)

7. What is the output of the following program? (6 points)

```
#include <iostream>
using namespace std;
template<class T>
void swapValues(T& v1, T& v2)
{
   T temp;
   temp = v1;
   v1 = v2;
   v2 = temp;
}
int main( )
{
   int first = 31, second = 54;
   double third = 8.7, fourth = 1.33;
   char fifth = 'z', sixth = 'a';
   swapValues(first, second);
   cout << first << " " << second << endl;
   swapValues(third, fourth);
   cout << third << " " << fourth << endl;
   swapValues(fifth, sixth);
   cout << fifth << " " << sixth << endl;
   return 0;
}
```


8. Which of the following statements are legal? (6 points)

```
class OneClass
{
public:
    OneClass(int newInfo, char moreNewInfo);
    OneClass( );
    void doStuff( );
private:
    int information;
    char moreInformation;
};
```

(a)　OneClass anObject(543, 'A');
(b)　OneClass otherObject;
(c)　OneClass anotherObject( );
(d)　anObject = OneClass(999, 'D');
(e)　anObject = OneClass( );
(f)　anObject = OneClass;

(a)	(b)	(c)	(d)	(e)	(f)

9. Which of the following statements are legal? (6 points)

```
class Car
{
public:
    void setPrice(double newPrice);
    void setProfit(double newProfit);
    double getPrice( );
private:
    double price;
    double profit;
    double getProfit( );
};
Car toyota, jaguar;
double aPrice, aProfit;

(a)     toyota.price = 19999;
(b)     jaguar.setPrice(25000);
(c)     aPrice = jaguar.getPrice( );
(d)     aProfit = jaguar.getProfit( );
(e)     if (toyota == jaguar) cout << "Want to swap cars?";
(f)     toyota = jaguar;
```

(a)	(b)	(c)	(d)	(e)	(f)

10. What is the output of the following program? (8 points)

```
#include <iostream>
#include <vector>
using namespace std;
int main( )
{
        vector<int> v(5);
        cout << "The size of v is " << v.size( ) << endl;
        v.push_back(3);
        v.push_back(4);
        v[0] = 1;
        v[1] = 2;
        cout << "The values of v are:" << endl;
        for (int j = 0; j < v.size( ); j++)
                cout << v[j] << "  ";
        cout << endl;
        cout << " The size of v is " << v.size( ) << endl;
        return 0;
}
```

11. Rewrite the following loop as a **for** loop. (6 points)

```
int i = 0;
while (i <= 100)
{
        if (i % 5 == 0) cout << 'X' << i << endl;
        i++;
}
```


12. Fill in the output in the following **Sample Dialogue**. (15 points)

```
#include <iostream>
using namespace std;
int main( )
{
        int *p1, *p2, *p3, *p4, n1, n2;
        n1 = 11;
        n2 = 22;
        p1 = &n1;
        p2 = &n2;
        p3 = &n1;
        p4 = &n2;
        p1 = p2;
        *p3 = *p4;
        n2 = 33;
        cout << &n1 << " " << &n2 << endl ;
        cout << p1 << " " << p2 << endl ;
        cout << p3 << " " << p4 << endl;
        cout << *p1 << " " << *p2 << endl ;
        cout << *p3 << " " << *p4 << endl;
        cout << &(*p1) << " " << &(*p3) << endl ;
        return 0;
}
```

Sample Dialogue

0	0	1	2	F	F	6	C		0	0	1	2	F	F	6	8		

13. The definition of the class **Money** is shown below. Implement the definition of the function **add**. (10 points)

```
class Money
{
public:
  friend Money add(const Money& amount1, const Money& amount2);
  // Precondition: amount1 and amount2 have been given values.
  // Returns the sum of the values of amount1 and amount2.
  Money(long dollars, int cents);
  Money(long dollars);
  Money( );
  double getValue( ) const;
  void input(istream& ins);
  void output(ostream& outs) const;
private:
  long allCents;
};
```

Money add(const Money& amount1, const Money& amount2)
{
}

14. What is the output of the following program? (3 points)

```
#include <iostream>
using namespace std;
int main( )
{
        int choice = 2;
        switch (choice + 1)
        case 1: cout << "Roast beef" << endl; break;
        case 2: cout << "Roast worms" << endl; break;
        case 3: cout << "Chocolate ice cream" << endl;
        case 4: cout << "Onion ice cream" << endl; break;
        default: "Delicious!" << endl;
        return 0;
}
```


15. The definition of the class **Money** is shown below. Implement the definition of the overloaded operator +. (10 points)

```
class Money
{
public:
   friend Money operator +(const Money& amount1, const Money& amount2);
   // Precondition: amount1 and amount2 have been given values.
   // Returns the sum of the values of amount1 and amount2.
   Money(long dollars, int cents);
   Money(long dollars);
   Money( );
   double getValue( ) const;
   void input(istream& ins);
   void output(ostream& outs) const;
private:
   long allCents;
};
```

Money operator +(const Money& amount1, const Money& amount2)
{
}

1. Fill in the output in the following **Sample Dialogue**? (6 points)

```
public class Array {
   public static void main(String[] args) {
        int[] array=new int[4];

        for (int i = 0; i < array.length; i++) {
           array[i]=(i<<2);
        }

        System.out.print(array[i]+" ");
   }
}
```

Sample Dialogue

0																	

2. Rewrite the following program so that it can compile and work correctly. (6 points)

```
public class Hello {
    private final String WELCOME="Welcome to the world of Java.";

    public int main() {
        System.out.println(Hello.WELCOME);
    }
}
```

public class Hello {
}

3. Fill in the output in the following **Sample Dialogue**. (4 points)

```
public class Iterator {
  private static int i=5;

  public static void main(String[] args) {
      System.out.println(i);

      for (int i=0; i<10; i++) {
          // do something;
      }

      System.out.println(i);
}
```

Sample Dialogue

5																									

4. While compiling the following code, we got a compiler error. Can you spot the problematic line about which the compiler complains? (6 points)

```
    class Dish {
(a)     private final String name;

(b)     private Dish(String name) {
            this.name=name;
        }

        public String toString() {
            return name;
        }

(c)     public static final Dish SALAD=new Dish("salad");
    }

(d) public class Orders extends Dish {
        // more stuff comes here.
    }
```


5. The partial definition of class Dump is given below. Please finish the implementation so that its execution result will be like what's shown in **Sample Dialogue.**(15 points)

```
public class Dump {
    public static void main(String[] args) {
        // file in your code here.
    }
}
```

Sample Dialogue

j	a	v	a		D	u	m	p		a		b		c	<Enter>						
a		**b**		**c**																	

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

6. Which of the follow statements about Java is true? (5 points)

(a) **Classes declared 'final' can be extended but their objects cannot be manipulated.**
(b) **Interfaces and abstract classes are almost identical. Actually, an interface is just an abstract class with all its methods declared 'abstract'. Thus, there are no true differences between an interface and an abstract class in Java.**
(c) **Objects declared 'final' cannot be manipulated.**
(d) **Static variables can only be accessed through static methods.**

(a)	(b)	(c)	(d)

7. What is the output of the following program? (6 points)

```
class A {
    protected String name() {
        return "A";
    }
}

class B extends A {
    protected String name() {
        return "B";
    }
}

class C extends B {
    protected String name() {
        return "C";
    }
}

public class Overriding {
    public static void main(String[] args) {
        A[] objs= new A[] {
            new A(),
            new B(),
            new C()
        };

        for (int i=0; i<objs.length; i++) {
            System.out.print(objs[i].name());
        }
    }
}
```

8. Which of the following statements are legal? (6 points)

```
public class OneClass {
    private int information;
    private char moreInformation;

    public OneClass();
    public OneClasss(int newInfo, char moreNewInfo);
    void doStuff( );
};
```

```
(a)     OneClass anObject(543, 'A');
(b)     OneClass otherObject;
(c)     OneClass anotherObject( );
(d)     anObject = OneClass(999, 'D');
```

(a)	(b)	(c)	(d)

9. There are compiling errors in the following program. Please point out which line can fix the problem. (6 points)

```
    public class Car {
(a)     private final int priceTag=0;

(b)     public Car(int price) {
            priceTag=price;
        }

        public int getPrice() {
            return priceTag;
        }

(c)     public static void main(String[] args) {
            Car car=new Car(100);

(d)         System.out.println("The price of the car is "+car.getPrice());
        }
    }
```

(a)	(b)	(c)	(d)

10. The partial definition of the class **DynamicArray** is shown below. Implement the definition of the function **dump** such that the execution of the program will dump out the content of the Vector v. (15 points)

```
import java.util.Vector;

public class DynamicArray {
  private Vector<Integer> v=new Vector<Integer>();

  public void add(int i) {
      v.add(new Integer(i));
  }

  public void dump() {
      // Your Implementation of dump() goes here.
  }

  public void static main(String[] args) {
       DynamicArray array=new DynamicArray();

       for (int i=0; i<10; i++) {
            array.add(i);
       }

       array.dump() ;
  }
}
```

Sample Dialogue

j	a	v	a		D	y	n	a	m	i	c	A	r	r	a	y	<Enter>		
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**										

```
public void dump() {

}
```

11. Rewrite the following loop as a **for** loop. (6 points)

```
int i = 0;
while (i <= 100) {
        if  (i % 5 == 0) {
            System.out.println(i);
        }

        i++;
}
```


12. Simplify the following nested **if**s into a **switch** statement. (6 points)

```
public char choice(int i) {
    char ch;

    if ((i%3)==0) {
        ch='A';
    }
    else if ((i%3) ==1) {
        ch='B';
    }
    else {
        ch='C';
    }

    return ch;
}
```

public char choice(int i) {
}

13. The partial definition of the class **Singleton** is shown below. Implement the definition of the function **getInstance** so that **TestSingleton** can work correctly. (10 points)

```
public class Singleton {
  public static Singleton INSTANCE=new Singleton();

  private Singleton() {
      // This constructor is left empty intentionally.
  }

  public void sayHello() {
      System.out.println("Hello, world!");
  }

  // Your implementation of getInstance() goes here.
  ...
}

public class TestSingleton{
  public static void main(String[] args) {
      Singleton obj=Singleton.getInstance();
      obj.sayHello();
  }
};
```

public static Singleton getInstance () {
}

14. What is the output of the following program? (3 points)

```
public class Switches {
    public static void main(String[] args) {
        int choice = 2;
        switch (choice + 1) {
            case 1:
                System.out.println("Roast beef, ");

            case 2:
                System.out.println("Roast worms, ");

            case 3:
                System.out.println("Chocolate ice cream and ");

            case 4:
                System.out.print("Onion ice cream ");

            default:
                System.out.println("are delicious");
        }
    }
}
```