

1. Which of the following determines where an identifier can be used in the text of a program?
- a) lifetime b) scope c) data type d) name

2. Consider the following statements

```
int *p= new int[100];  
delete []p;
```

Which of the following describes best what the delete statement does?

- a) allocates memory
b) deletes the pointer
c) de-allocates memory used for the 0'th element of the array
d) de-allocates memory used for the entire array

[3-6]. The next four questions refer to the Circuit class defined below:

```
#include <iostream.h>  
const int PARALLEL=0;  
const int SERIES  =1;  
  
class Circuit {  
private:  
    double voltage;  
    double res1;  
    double res2;  
    int type;  
public:  
    Circuit();  
    Circuit(double v, double r1, double r2, int t);  
    double getVoltage(void);  
    void setVoltage(double v);  
    void setResistance(double r1, double r2, int t);  
    double findCurrent(void);  
    double getTotalResistance(void);  
};  
  
Circuit::Circuit() {  
    voltage=res1=res2=0;  
    type = PARALLEL;  
}  
  
Circuit::Circuit(double v, double r1, double r2, int t){  
    voltage = v;  
    res1 = r1;  
    res2 = r2;  
    type = t;  
}  
  
double Circuit::getVoltage(void)
```

```
{  
    return voltage;  
}  
void Circuit::setVoltage(double v)  
{  
    voltage = v;  
}  
void Circuit::setResistance(double r1,  
    double r2, int t){  
    res1 = r1;  
    res2 = r2;  
    type = t;  
}  
  
double Circuit::findCurrent(void) {  
    if (getTotalResistance()==0) {  
        return 0;  
    }  
    else {  
        return voltage / getTotalResistance();  
    }  
}  
  
double Circuit::getTotalResistance(void){  
    if (type == SERIES) {  
        return (res1+res2);  
    }  
    else if (type == PARALLEL) {  
        return (res1*res2)/(res1 + res2);  
    }  
    else {  
        return 0;  
    }  
}
```

3. Consider the following statement:

```
Circuit c(5.8,280,370,1);
cout << c.getTotalResistance() << ' ' << c.getVoltage();
```

What is the output?

- a). 650 5.8 b). 159 5.8 c). 5.8 650 d). 5.8 159

4. Consider the following program segment

```
Circuit c1;
Circuit c2(1200, 2, 2, 1);
cout << c1.getVoltage() << ',';
cout << c2.getVoltage() << ',';
c1.setResistance(2,2,0);
cout << c1.findCurrent() << ',';
cout << c2.findCurrent();
```

What is the output?

- a). 0,1200,0,1200 b). 0,0,0,0 c). 0,1200,0,300 d). 1200,1200,0,300

5. Consider the following statement:

```
Circuit c1(2400, 4, 4, 0);
Circuit c2(1200, 2, 2, 1);
c2.setVoltage(c1.getVoltage());
cout << c1.getVoltage() << ',';
cout << c2.getVoltage() << endl;
```

What is the output?

- a). 1200,1200 b). 2400,1200 c). 1200,2400 d). 2400,2400

6. Consider the following program segment

```
Circuit c(1200, 2, 2, 1);
c.setResistance(5,5,0);
cout << c.findCurrent() << endl;
```

What is the output?

- a). 120 b). 300 c). 1200 d). 480

7. Consider the following C++ program:

```
#include <iostream>
int f1(int &a, int &b, int c[]);
int f2(int i1, int i2);
int main(void) {
    int x = 2, y = 3, z[] = {2, 6, 2, 4, 8};
    cout << f1(x, y, z) << endl;
    return 0;
}
```

```
int f1(int &a, int &b, int c[]) {
    return f2(c[a], c[b]);
}

int f2(int i1, int i2) {
    return (i1 + i2) / 2;
}
```

Which of the following is the resulting output?

- a). 2 b). 3 c). 4 d). 5

8. What is the value of x after the following statements have been executed?

```
int x, y = 14, z = 5;
```

x = static_cast<double>(y)/ z;

- a). 2 b). 2.8 c). 4 d). 0

[9-13]. Read the following program and answer the next questions.

```
#include <iostream.h>
```

```
class A
```

```
{
```

```
    int i;
```

```
public:
```

```
    A(){ cout<<'A'; }
```

```
    virtual void f() { cout<<"FA";};
```

```
};
```

```
class B: virtual public A
```

```
{
```

```
    int i;
```

```
public:
```

```
    B(){ cout<<'B'; }
```

```
    void f() { cout<<"FB";};
```

```
};
```

```
class C: virtual public A
```

```
{
```

```
    int i;
```

```
public:
```

```
    C(){ cout<<'C'; }
```

```
    void f() { cout<<"FC";};
```

```
};
```

```
class D: public B, public C
```

```
{
```

```
    int i;
```

```
public:
```

```
    D(){ cout<<'D'; }
```

```
    void f() { cout<<"FD";};
```

```
};
```

9. Consider the following statement:

```
D d;
```

What is the output?

- a). ABACD b). BACAD c). ABCD d). DBCA

10. Consider the following program segment

```
A* p = &d;    // d is an object of D
```

```
p->f();
```

What is the output?

- a). FA b). FB c). FC d). FD

11. If we delete all keywords “virtual”, sizeof(D) should be ____.

- a). 28 b). 24 c). 20 d). 16

Consider the following program segment

```
L1   D d;
```

```
L2   A a;
```

```
L3   A *pa = &a;
```

```
L4   D *pd = &d;
```

```
L5   B* pb = dynamic_cast<B*>(pd) ;
```

```
L6   pd = dynamic_cast<D*>(pb);
```

```
L7   pa = pd;
```

```
L8   pd = pa;
```

```
L9   a = d;
```

```
L10   d = a;
```

12. Which lines are compiler errors?

- a). L6&L10 b). L6&L8&L10 c). L8&L10 d). L9&L10

13. Which line(s) is compiler warning?

- a). L5&L6 b). L6 c). L6&L8 d). L5

14. In C++, the term “default parameter” means that _____.
 a) C++ supplies default functionality you get for free.
 b) A parameter that C++ automatically supplies a default value for.
 c) The function parameters are designed when they are defined.
 d) A parameter based on primitive C++ variable types.
15. _____ is right.
 a) `char * const p = "aa"; p[0]='b';`
 b) `const char * const p = "aa"; p[0]='b';`
 c) `const char *p = "aa"; p[0]='b';`
 d) `char const *p = "aa"; p[0]='b';`

Part II. Fill in the blanks. (36 points)

1. What is the output of the following program: (11 points)

```
#include <iostream>
using namespace std;
class Sample
{
public:
    Sample()    { j += 5;  cout << "j = " << j << endl;  }
    void print() { cout << "j = " << j << endl;          }
    ~Sample()   { j -= 3;  cout << "j = " << j << endl;  }
private:
    static int j;
};
int Sample::j = 20;
void main()
{
    Sample S1, S2;
    S2.print();
}
```

Here gives the outputs:

2. Consider the following function definition. (4 points)

```
void Twist(int a, int& b)
{
    a = a*2;
    b = b + 2;
}
```

Show the exact output for each successive cout statement on the line provided.

```
int s=2, t=3;
Twist(s,t);
cout << s << "    " << t;
Twist(t,s);
cout << s << "    " << t;
```

Here gives the outputs:

3. Consider the following function definition.(9 points)

```
#include <iostream>
using namespace std;
class Sample
{
public:
    Sample() { cout << "Default Constructor is called." << endl; }
    Sample(const Sample& S){ cout << "Copy onstructor is called." << endl; }
    ~Sample() { cout << "Destructor is called." << endl; }
};
Sample Fun() { Sample S; return S;}
void main()
{
    Sample S = Fun();
}
```

Here gives the outputs:

4. Suppose we have a templated class **Jedi** whose private variables are a dynamic array list of type T and an integer size that tracks the number of elements in the array. The declaration for the **Jedi** class and its default constructor are shown below.(12 points)

```
template <typename T>
class Jedi {
public:
    Jedi();
    ...
private:
    T* list;
    int size;
```

```
};

template <typename T>
Jedi<T>::Jedi() {
    list = NULL;
    size = 0;
}
```

Finish the definitions for the other 3 member functions:

```
_____
Jedi<T>::Jedi (_____Jedi<T>& right) {
    size = right.size;
    list = new T[size];
    for (int i=0; i<size; i++)
        list[i] = right.list[i];
}
Jedi<T>& Jedi<T>::operator= (_____ right) {
    if (this != &right) {
        delete[] list;
        size = right.size;
        _____;
        for (int i=0; i<size; i++)
            list[i] = right.list[i];
    }
}
```

```

    }
    return _____;
}
template <typename T>
Jedi<T>::~Jedi () { _____; }

```

PartIII. Programming. (34 points)

1. (14 points) Define a class, **Point**, with demands as follows:

- (1) The class has member data **X** and **Y** which describe a point's coordinates.
- (2) Define a member function **Distance** to get the distance(距离) between two points.
- (3) Define a friend function **Distance** to get the distance between two points.
- (4) A programmer can use **Point** in main() as follows:

```

void main()
{
    Point p, q(2, 3); // The coordinate of point p is original point.
    cout << p.Distance(q) << endl;
    cout << Distance(p, q) << endl;
}

```

2. (20 points) **Real** is combined with numerator(分子) and denominator(分母). Define a class, **Real**, with demands as follows:

- (1) The class has both **numerator** and **denominator** initialized as member data in the constructor.
- (2) Define association functions and programmer can use **Real** in **main()** as follows:

```

void main(void)
{
    Real R1(1, 2), R2(2, 3), R3; // R3 is zero with default.
    R3 = R1 + R2;
    double number = R3;
    cout << R3 << endl;
    cout << number << endl;
}

```