

Programming in C++: Assignment Week 6

Total Marks : 20

Partha Pratim Das
Department of Computer Science and Engineering
Indian Institute of Technology
Kharagpur – 721302
partha.p.das@gmail.com

April 6, 2017

Question 1

Look at the code snippet below and find out What will be the correct syntax for up-casting ?

Mark 1

```
class Parent {
public:
    void sleep() {}
};

class Child: public Parent {
public:
    void gotoSchool(){}
};

int main(){
    Parent parent;
    Child child;
    return 0;
}
```

- a) Child *pChild = (Child *) &parent;
- b) Child *pChild = &parent;
- c) Parent *pParent = &child;
- d) Parent *pParent = (Parent *) &child;

Answer: c)

Explanation: By definition of upcasting (Base Class to Derive class)

Question 2

What will be the correct syntax for down-casting in above code (QS No-1)?

- a) Child *pChild = (Child *) &parent;

- b) Child *pChild = &parent;
- c) Parent *pParent = &child;
- d) Parent *pParent = (Parent *) &child;

Answer: a)

Explanation: By definition of downcasting (Derive class to Base Class)

Question 3

Look at the following code snippet and Find out the static binding from the option below

Mark 1

```
class Base {
public:
    void first() {}
    virtual void second() {}
};
class Derived : public Base {
public:
    void f() {}
    virtual void g() {}
};

int main() {
    Base b;
    Derived d;
    Base *pb = &b;
    Base *pd = &d;
    Base &rb = b;
    Base &rd = d;

    pb->first(); // 1
    pd->first(); // 2
    rb.first(); // 3
    rd.first(); // 4
    pb->second(); // 5
    pd->second(); // 6
    rb.second(); // 7
    rd.second(); // 8
}
```

- a) 1 2 5 6
- b) 3 4 7 8
- c) 1 2 3 4
- d) 5 6 7 8

Answer: c)

Explanation: Direct function calls can be resolved using a process known as early binding or static binding.

Question 4

Consider the code of question 3 and find out the dynamic binding from the option below

- a) 1 2 5 6
- b) 3 4 7 8
- c) 1 2 3 4
- d) 5 6 7 8

Answer: d)

Explanation: Virtual functions get resolve at run time.

Question 5

Consider the following code snippet. Find out the correct sequence of function calling. *Marks: 1*

```
class X {
public:
    virtual void f() { } //1
    void g() { } //2
};

class Y : public X {
public:
    void f() { } //3
    virtual void g() { } //4
};

int main() {
    X x; Y y;

    X& rx = y;

    x.f();
    x.g();

    rx.f();
    rx.g();

    return 0;
}
```

- a) 1 2 3 2
- b) 1 2 3 4
- c) 3 2 1 2
- d) 3 2 1 4

Answer: a)

Explanation: rx.f() call Y::f as f is virtual in the base class.

Question 6

What will be the output/Error of the bellow code snippet?

```
class Base {
public:
virtual void show() = 0;
};

class Derived : public Base{
public:
void show(){ cout << "Virtual Function";}
};

int main()
{
Base obj;
Base *b;
Derived d;
b = &d;
b->show();
}
```

- a) Compile Time Error: cannot create object
- b) Compile Time Error: cannot inherit Base class
- c) Compile Time Error: cannot overload show
- d) Virtual Function

Answer: a)

Explanation: Can't create of object of a abstract base class

Question 7

What will be the output of the following program?

Marks: 1

```
#include<iostream>
using namespace std;

class base {
public:
    base() { cout << "c"; }
    virtual ~base() { cout << "~c"; }
};

class derived : public base {
public:
    derived() { cout << "d"; }
    ~derived() { cout << "~d"; }
};

int main(void)
```

```

{
    {
        derived *d = new derived();
        base *b = d;
        delete b;
    }
    return 0;
}

```

- a) $d \sim c \sim d$
- b) $d \sim c \sim d \sim c$
- c) $c \sim d \sim c \sim d$
- d) $c \sim d \sim d \sim c$

Answer: d)

Explanation: Deleting a derived class object using a pointer to a base class that has a non-virtual destructor results in undefined

Question 8

Find the abstract classes from the bellow code?

```

class Instrument {
public:
    virtual void play() = 0
    { cout << "Instrument: Init Brush" << endl; }
};
class Wind : public Instrument {
    void play() { cout << "Polygon: play" << endl; }
};
class Percussion : public Instrument {
};
class Woodwind : public Wind {
public:
    void play() { cout << "Woodwind: play" << endl; }
};
class Brass : public Wind {
public:
    void play() { cout << "Brass: play" << endl; }
};
class Drum : public Percussion {
public:
    void play() { cout << "Drum: play" << endl; }
};
class Tambourine : public Percussion {
public:
    void play() { cout << "Tambourine: play" << endl; }
};

```

- a) Woodwind, Brass

- b) Instrument, Percussion
- c) Instrument, Tambourine
- d) Percussion, Drum

Answer: b)

Explanation: pure virtual function play() is there in Instrument and Percussion class

Question 9

What will be the output of the following Code Snippet ?

Marks: 1

```
class Instrument {
public:
    virtual void play()
    { cout << "1 "; }
};
class Wind : public Instrument {
    void draw() { Instrument::play(); cout << "2 "; }
};
class Percussion : public Instrument {
};
class Woodwind : public Wind {
public:
    void play() { Instrument::play(); cout << "3 "; }
};
class Brass : public Wind {
public:
    void play() { Instrument::play(); cout << "4 "; }
};
class Drum : public Percussion {
public:
    void play() { Instrument::play(); cout << "5 "; }
};
class Tambourine : public Percussion {
public:
    void play() { Instrument::play(); cout << "6 "; }
};
int main() {
    Instrument *arr[] = { new Woodwind, new Brass, new Drum, new Tambourine };
    for (int i = 0; i < sizeof(arr) / sizeof(Instrument *); ++i) arr[i]->play();

    return 0;
}
```

- a) 3 4 5 6
- b) 1 3 1 4 1 5 1 6
- c) 1 3 4 5 6
- d) Compile Time Error: Virtual function can't have body

Answer: b)

Explanation:: Virtual function may have body

Question 10

What will be the output of the following code snippet?

Marks: 1

```
class A { public: int i; };  
class B { public: double d; };
```

```
A a;  
B b;  
a.i = 8;  
b.d = 9.7;  
A *p = &a;  
B *q = &b;
```

```
p = (A*)&b;  
q = (B*)&a;  
cout << p->i << endl;  
cout << q->d << endl;
```

- a) 9.7 8
- b) 8 9.7
- c) GARBAGE
- d) Compile Time Error: casting is not possible

Answer: c)

Explanation: Force casting between unrelated classes

Programming Questions

Question 1

Problem Statement:

Marks: 2

Consider the following code. Modify the code in editable section to match the public test cases.

```
#include <iostream>  
using namespace std;  
  
class B {  
public:  
    B() { cout << "98 "; } // don't modify the "cout"  
  
    ~B() { cout << "56 "; } // Don't Edit/Modify the "cout"  
  
};  
  
class D : public B {
```

```

        int n;
public:
    D(int p):n(p) { cout << n << " "; }
    ~D() { cout << n*2 << " "; }
};

int main() {
    int n ; cin >> n ;

    B *basePtr = new D(n);

    delete basePtr;

    return 0;
}
//-----

```

Public-1

Input:
2
Output:
98 2 4 56

Public-2

Input:
8
Output:
98 8 16 56

Private

Input:
3
Output:
98 3 6 56

Answer: virtual

Question 2

Consider the skeletal code below. When the program is executed, the member functions (excluding the constructors and destructors) are called in the order: A::g B::g B::f A::g B::g C::g C::f. Fill up the blanks to match the test cases. *Marks: 2*

```

#include <iostream>
using namespace std;

class A { protected: int ai;
public:
    A(int i) : ai(i) {}
    _____ void f() = 0;           // Fill the blank or Remove blank
    _____ void g()               // Fill the blank or Remove blank
    { ++ai; }

```



```

};
class B : public A { protected: int bi;
public:
    B(int i) : A(i), bi(i) {}
    ----- void f()                // Fill the blank or Remove blank
    { cout << ai << bi; }          // DO NOT EDIT THIS LINE

    ----- void g()                // Fill the blank or Remove blank
    { A::g(); }
};
class C : public B { int ci;
public:
    C(int i) : B(i), ci(i) {}
    ----- void f()                //Fill the blank or Remove blank
    { cout << ai << bi << ci; } // DO NOT EDIT THIS LINE

    ----- void g()                //Fill the blank or Remove blank
    { B::g(); }
};
int main() {
    int x = 3 ;
    int y;
    cin >> y;

    A *p[] = { new B(x), new C(y) };
    for(int i = 0; i < sizeof(p) / sizeof(A*); ++i)
        { p[i]->g(); p[i]->f(); }
    return 0;
}

```

Further, the input / output test case is as follows:

Public 1

Input: 2

Output: 43322

Public 2

Input: 23

Output: 43242323

Private

Input: 4

Output: 43544

Answer: virtual // virtual // virtual or blank // virtual or blank // virtual or blank
 // virtual or blank

Question 3

Consider the following code. write the proper definition of "getArea()" in the editable section so that the test cases would pass .

Marks: 2

```

#include <iostream>
using namespace std;

class Shape {
protected:
    int width, height;
public:

    // Write the getArea() function here

    void setWidth(int w) { width = w; }
    void setHeight(int h) { height = h; }
};

class Rectangle : public Shape {
public:
    int getArea() { return (width * height); }
};

class Triangle : public Shape {
public:
    int getArea() { return (width * height) / 2; }
};

int main(void) {
    int x, y;
    cin >> y >> x;
    Rectangle Rect;
    Triangle Tri;

    Rect.setWidth(x);
    Rect.setHeight(y);

    Tri.setWidth(x);
    Tri.setHeight(y);

    Shape *shape[] = { &Rect, &Tri, 0 };
    Shape **pShape = &shape[0];

    while (*pShape)
        cout << (*pShape++)->getArea() << " ";

    return 0;
}

```

Public-1

Input:

3

7

Output:

21 10

Public-2

Input:

35

23

Output:

805 402

Private

Input:

6

9

Output:

54 27

Answer:

```
virtual int getArea() = 0;
```

Question 4

Consider the following code. Fill up the code in editable section to congratulate the Manager and the Clerk, so that outputs will be matched as given in the test cases. *Marks: 2*

```
#include <iostream>
#include <string>
using namespace std;

class Employee {
public:
    string Name;
    double salary;
    Employee(string fName, double sal) : Name(fName), salary(sal) {}
    void show() {
        cout << Name << " " << salary;
    }
    void addBonus(double bonus) {
        salary += bonus;
    }
};

class Manager :public Employee {
public:
    Manager(string fName, double sal) : Employee(fName, sal) {}
};
```

```

class Clerk :public Employee {
public:
    Clerk(string fName, double sal) : Employee(fName, sal) {}
};

void congratulate(Employee* emp) {
    emp->addBonus(200);
    emp->show();
    cout << " ";
};

int main() {
    Employee* emp;
    int sal_m, sal_c;
    cin >> sal_c >> sal_m;
    Manager m1("Steve", sal_m);
    Clerk c1("Kevin", sal_c);

    // Call the proper function to congratulate the Manager and the Clerk

    return 0;
}

```

Public-1

Input:
2000
1000
Output:
Kevin 2200 Steve 1200

Public-2

Input:
4000
1000
Output:
Kevin 4200 Steve 1200

Private

Input:
6200
2500
Output:
Kevin 6400 Steve 2700

Answer:

```

congratulate(&c1);
congratulate(&m1);

```

Question 5

```
#include <iostream>
using namespace std;

class Shape {
protected:
    int width, height;

public:
    Shape(int a = 0, int b = 0) {
        width = a;
        height = b;
    }
    // Fill the blank to run the code
    ----- void area() {
        cout << "Parent class area :" << endl;
    }
};

class Rectangle : public Shape {
public:
    Rectangle(int a = 0, int b = 0) :Shape(a, b) { }
    void area() {
        cout << (width * height) << " ";
    }
};

class Triangle : public Shape{
public:
    Triangle(int a = 0, int b = 0) :Shape(a, b) { }
    void area() {
        cout << (width * height / 2);
    }
};

// Main function for the program
int main() {
    int a, b;
    cin >> a;
    cin >> b;
    Shape *shape;
    Rectangle rec(a, b);
    Triangle tri(a, b);

    // Write the code to call rectangle area.
    -----
    shape->area();

    // Write the code to call triangle area.
    -----
```

```
        shape->area();  
  
        return 0;  
    }
```

Public-1

Input:
10 5
Output:
50 25

Public-2

Input:
10 25
Output:
250 125

Private

Input:
60 30
Output:
1800 900

Answer:

```
virtual  
shape = &rec;  
shape = &tri;
```