

कंस्ट्रक्टर और डिस्ट्रक्टर

पाठ्यपुस्तक के प्रश्न

वस्तुनिष्ठ प्रश्न

प्रश्न 1. इनमें से कौनसा कंस्ट्रक्टर के संदर्भ में सत्य है?

- (अ) इसका नाम क्लास के नाम जैसा ही होता है।
- (ब) ये क्लास के पब्लिक अनुभाग में घोषित होने चाहिए।
- (स) जब ऑब्जेक्ट घोषित किये जाते हैं तब ये स्वतः ही कॉल हो जाते हैं।
- (द) उपरोक्त सभी

उत्तर: (द) उपरोक्त सभी

प्रश्न 2. कंस्ट्रक्टर जो आरग्यूमेन्ट लेते हैं उन्हें क्या कहा जाता है?

- (अ) डिफॉल्ट कंस्ट्रक्टर
- (ब) बिना आरग्यूमेन्ट का
- (स) पैरामीटराइज्ड कंस्ट्रक्टर
- (द) इनमें से कोई नहीं

उत्तर: (स) पैरामीटराइज्ड कंस्ट्रक्टर

प्रश्न 3. कंस्ट्रक्टर जो एक ऑब्जेक्ट को उसी क्लास के दूसरे ऑब्जेक्ट से घोषित और इनिशियलाइज करता है, उसे कहा जाता है

- (अ) डिफॉल्ट कंस्ट्रक्टर
- (ब) कॉपी कंस्ट्रक्टर
- (स) पैरामीटराइज्ड कंस्ट्रक्टर
- (द) इनमें से कोई नहीं

उत्तर: (ब) कॉपी कंस्ट्रक्टर

प्रश्न 4. इनमें से कौनसा डिस्ट्रक्टर के संदर्भ में सत्य है?

- (अ) इसका नाम क्लास के नाम के जैसा होता है। लेकिन इसके नाम से पहले tidle (~) चिह्न होता है।
- (ब) यह आरग्यूमेन्ट नहीं लेता है और वेल्यू भी रिटर्न नहीं करता है।
- (स) यह कम्पाइलर के द्वारा स्वतः ही कॉल होता है जब किसी प्रोग्राम या ब्लॉक या फंक्शन से बाहर आते हैं।
- (द) उपरोक्त सभी

उत्तर: (द) उपरोक्त सभी

अतिलघु उत्तरीय प्रश्न

प्रश्न 1. कंस्ट्रक्टर क्या होते हैं?

उत्तर- कंस्ट्रक्टर एक विशेष प्रकार के मेम्बर फंक्शन हैं। ये क्लास के ऑब्जेक्ट को इनिशियलाइज (initialize) करने के लिए काम आते हैं। इनका नाम क्लास के नाम जैसा ही होता है।

प्रश्न 2. पैरामीटराइज्ड कंस्ट्रक्टर क्या होते हैं?

उत्तर- जो कंस्ट्रक्टर आरग्यूमेन्ट लेते हैं, उन्हें पैरामीटराइज्ड कंस्ट्रक्टर कहा जाता है। ऑब्जेक्ट create करते समय हम parameterized constructors को arguments pass कर सकते हैं जो कि class variables को assign किए जा सकते हैं।

प्रश्न 3. कंस्ट्रक्टर ओवरलोडिंग किसे कहते हैं?

उत्तर- एक क्लास में एक से ज्यादा कंस्ट्रक्टर हो सकते हैं और इसे कंस्ट्रक्टर ओवरलोडिंग कहा जाता है। एक ही नाम के कंस्ट्रक्टर अलग-अलग पैरामीटर (parameters) के साथ ही क्लास में declare किये जा सकते हैं। यही कंस्ट्रक्टर ओवरलोडिंग है।

लघु उत्तरीय प्रश्न

प्रश्न 1. कंस्ट्रक्टर की क्या विशेषताएँ होती हैं?

उत्तर- कंस्ट्रक्टर की निम्नलिखित विशेषताएँ होती हैं

- इसका नाम क्लास के नाम जैसा ही होता है।
- ये क्लास के पब्लिक अनुभाग में घोषित होने चाहिए।
- जब ऑब्जेक्ट घोषित किया जाते हैं तब ये स्वतः ही कॉल हो जाते हैं।
- इनका कोई रिटर्न टाइप नहीं होता है, यहाँ तब की void भी नहीं, और इस प्रकार ये कोई वेल्फू रिटर्न नहीं करते हैं।
- इनको इनहेरिट नहीं किया जा सकता है।
- इनके एड्रेस को एक्सेस नहीं कर सकते हैं।

प्रश्न 2: डिस्ट्रक्टर क्या होते हैं? इनकी विशेषताएँ लिखो।

उत्तर- डिस्ट्रक्टर-क्लास का एक विशेष प्रकार का मेंबर फंक्शन जिसका प्रयोग कंस्ट्रक्टर के द्वारा बनाये गये ऑब्जेक्ट को खत्म करने के लिए किया जाता है डिस्ट्रक्टर कहलाता है। डिस्ट्रक्टर फंक्शन की निम्नलिखित विशेषताएँ होती हैं

- इसका नाम क्लास के नाम के जैसा होता है लेकिन इसके नाम से पहले tilde (~) चिह्न होता है।
- यह आरग्यूमेन्ट नहीं लेता है और वेल्यू भी रिटर्न नहीं करता है।
- जब किसी प्रोग्राम या ब्लॉक या फंक्शन से बाहर आते हैं तब यह कम्पाइलर के द्वारा स्वतः ही कॉल होता है।

प्रश्न 3. डिस्ट्रक्टर के उपयोग लिखो।

उत्तर- डिस्ट्रक्टर का उपयोग ऑब्जेक्ट को आंशिकतः मैमोरी को रन टाइम पर मुक्त करने के लिए किया जाता है। ताकि मुक्त मैमोरी का पुनः उपयोग दूसरे प्रोग्राम या ऑब्जेक्ट के लिए किया जा सके। किसी ऑब्जेक्ट को मैमोरी का आवंटन कंस्ट्रक्टर फंक्शन में new ऑपरेटर से किया जाता है और डिस्ट्रक्टर फंक्शन में delete ऑपरेटर से आवंटित मैमोरी को पुनः लिया जाता है।

निबंधात्मक प्रश्न

प्रश्न 1. डिफॉल्ट आरग्यूमेन्ट के साथ कंस्ट्रक्टर का वर्णन कीजिए।

उत्तर- डिफॉल्ट आरग्यूमेन्ट के साथ कंस्ट्रक्टर
कंस्ट्रक्टर आरग्यूमेन्ट ले सकते हैं।

उदाहरण के लिए-

```
point (int a, int b=0);
```

ध्यान दें कि डिफॉल्ट आरग्यूमेन्ट दायें से बायें दिये जाते हैं। आरग्यूमेन्ट b की डिफॉल्ट वेल्यू शून्य है। तब यह स्टेटमेंट

```
point (5);
```

a को 5 वेल्यू असाइन करता है और b को शून्य (डिफॉल्ट द्वारा) लेकिन स्टेटमेंट

```
point(7, 11);
```

a को 7 वेल्यू असाइन करता है और b को 11 क्योंकि जब वास्तविक पैरामीटर दिए गए हों तब वे डिफॉल्ट आरग्यूमेन्ट को ओवरराइड कर देते हैं।

अगर एक आरग्यूमेन्ट वाला कंस्ट्रक्टर भी इस कंस्ट्रक्टर के साथ मौजूद है तब कॉल करने वाला स्टेटमेंट

```
point (5);
```

यह फैसला लेने में असमर्थ है कि कौन से कंस्ट्रक्टर को कॉल किया जाये और एक अस्पष्टता की स्थिति उत्पन्न हो जाती है। कम्पाइलर एक त्रुटि संदेश उत्पन्न कर देता है।

अन्य महत्वपूर्ण प्रश्न

अतिलघु उत्तरीय प्रश्न

प्रश्न 1. कंस्ट्रक्टर का रिटर्न टाईप क्या होता है?

उत्तर- कंस्ट्रक्टर का कोई रिटर्न टाईप नहीं होता है, यहाँ तक की void भी नहीं और इस प्रकार ये कोई वेल्यू रिटर्न नहीं करता है।

प्रश्न 2. क्या कंस्ट्रक्टर को इनहेरिट किया जा सकता है?

उत्तर- नहीं, कंस्ट्रक्टर को इनहेरिट नहीं किया जा सकता है।

प्रश्न 3. डिफॉल्ट कंस्ट्रक्टर किसे कहते हैं?

उत्तर- जो कंस्ट्रक्टर आरग्यूमेन्ट नहीं लेते हैं उन्हें डिफॉल्ट कंस्ट्रक्टर कहा जाता है।

प्रश्न 4. पैरामीटराइज्ड कंस्ट्रक्टर को कितने तरीकों से कॉल किया जा सकता है?

उत्तर- पैरामीटराइज्ड कंस्ट्रक्टर को दो तरीकों से कॉल किया जा सकता है

- Explicit call
- Implicit call

प्रश्न 5. कंस्ट्रक्टर फंक्शन को अन्य किस रूप में परिभाषित किया जा सकता है?

उत्तर- कंस्ट्रक्टर फंक्शन को इनलाईन फंक्शन के रूप में भी परिभाषित किया जा सकता है।

प्रश्न 6. क्या एक कंस्ट्रक्टर अपनी क्लास का रेफरेंस एक ऑरग्यूमेन्ट के रूप में ले सकता है?

उत्तर- हाँ, एक कंस्ट्रक्टर अपनी क्लास का रेफरेंस एक आरग्यूमेन्ट के रूप में ले सकता है।

प्रश्न 7. डाइनामिक इनिशियलाइजेशन क्या होता है?

उत्तर- एक ऑब्जेक्ट की इनिशियल वेल्यू रन टाइम पर दिया जाना ही डाइनामिक इनिशियलाइजेशन कहलाता है।

प्रश्न 8. डाइनामिक इनिशियलाइजेशन का क्या फायदा है?

उत्तर- डाइनामिक इनिशियलाइजेशन का फायदा यह है कि हम विभिन्न प्रकार के इनपुट कंस्ट्रक्टर ओवरलोडिंग के द्वारा दे सकते हैं।

प्रश्न 9. कॉपी कंस्ट्रक्टर से आप क्या समझते हो?

उत्तर- कंस्ट्रक्टर जो एक ऑब्जेक्ट को उसी क्लास के दूसरे ऑब्जेक्ट से घोषित और इनिशियलाइज करता है, उसे कॉपी कंस्ट्रक्टर कहा जाता है। एक कॉपी कंस्ट्रक्टर उसी क्लास के ऑब्जेक्ट का रेफरेंस आरग्यूमेन्ट के रूप में लेते हैं।

प्रश्न 10. डिस्ट्रक्टर की एक विशेषता बताइए।

उत्तर- इसका नाम क्लास के नाम जैसा होता है। लेकिन इसके नाम से पहले tilde (~) चिह्न होता है।

प्रश्न 11. डिस्ट्रक्टर फंक्शन में आवंटित मेमोरी को पुनः वापिस कैसे लिया जाता है?

उत्तर- डिस्ट्रक्टर फंक्शन में delete ऑपरेटर से आवंटित मेमोरी को पुनः लिया जाता है।

लघु उत्तरीय प्रश्न

प्रश्न 1. कंस्ट्रक्टर का एक उदाहरण दीजिए।

उत्तर-

```
class point
{
int x,y;
public:
point (void);//constructor declared
.....
.....
};
point::point (void)//constructor defined
{
x = 0;
y = 0;
}
```

जब हम क्लास point के ऑब्जेक्ट को घोषित करते हैं
उदाहरण के लिए-

```
point p;
```

क्लास के अन्दर कंस्ट्रक्टर स्वतः ही कॉल हो जाता है और प्राइवेट डेटा मेंबर x और y को शून्य से ऑब्जेक्ट p के लिए इनिशियलाइज कर देते हैं। जो कंस्ट्रक्टर आरग्यूमेन्ट नहीं लेते हैं उसे डिफॉल्ट कंस्ट्रक्टर कहा जाता है। अगर ऐसा कोई कंस्ट्रक्टर क्लास में परिभाषित नहीं है तब कम्पाइलर डिफॉल्ट कंस्ट्रक्टर उपलब्ध करवाता है उस क्लास के ऑब्जेक्ट घोषित करने के लिए।

प्रश्न 2. पैरामीटराइज्ड कंस्ट्रक्टर को उदाहरण सहित समझाइए।

उत्तर- पैरामीटराइज्ड कंस्ट्रक्टर

जो कंस्ट्रक्टर आरग्यूमेन्ट लेते हैं उन्हें पैरामीटराइज्ड कंस्ट्रक्टर कहा जाता है।
उदाहरण के लिए-

```
class point
{
int x, y;
public:
point (int a, int b);//parameterized constructor
{
.....
.....
}
};
point::point(int a, int b)
{
x=a;
y=b;
}
```

पैरामीटराइज्ड कंस्ट्रक्टर को दो तरीकों से कॉल किया जा सकता है।

```
point p = point (10, 20); //explicit call
```

यह स्टेटमेन्ट एक ऑब्जेक्ट p को 10 और 20 वैल्यू से इनिशियलाइज करता है।

```
point p (10, 20); //implicit call
```

यह स्टेटमेन्ट उपरोक्त स्टेटमेन्ट की तरह काम करता है।

प्रश्न 3. कंस्ट्रक्टर फंक्शन को इनलाईन फंक्शन के रूप में परिभाषित करने का एक उदाहरण दीजिए।

अथवा

इनलाईन फंक्शन का एक प्रोग्राम लिखिए।

उत्तर- कंस्ट्रक्टर फंक्शन को इनलाईन फंक्शन के रूप में भी परिभाषित किया जा सकता है।
उदाहरण के लिए

```
class point
{
int x,y;
public:
point (int a, int b)
{
x=a;
y=b;
}
.....
.....
};
```

कंस्ट्रक्टर के ऑरग्यूमेन्ट किसी भी टाईप के हो सकते हैं सिवाय उस क्लास के जिससे ये सम्बन्धित हैं।

प्रश्न 4. डिस्ट्रक्टर फंक्शन का एक प्रोग्राम लिखिए।

अथवा

डिस्ट्रक्टर फंक्शन कम्पाइलर के द्वारा स्वतः ही कॉल होता है, प्रोग्राम द्वारा दर्शाइए।

उत्तर- प्रोग्राम-डिस्ट्रक्टर फंक्शन

```
#include<iostream>
using namespace std;
class sample
{
sample () // Construtor
{
cout<<"Object created\n";
}
~sample () //Destructor
{
cout<< "Object destroyed";
}
}
```

```
};
int main ().
{
sample s;
return 0;
}
```

प्रोग्राम का आउटपुट होगा
Object created
Object destroyed

प्रश्न 5. डिस्ट्रक्टर के उपयोग से मैमोरी को मुक्त करने के लिए प्रोग्राम लिखिए।

उत्तर- प्रोग्राम-डिस्ट्रक्टर का उपयोग मैमोरी को मुक्त करने के लिए

```
#include<iostream>
using namespace std;
class sample
{
char *t;
public:
sample (int length)
{
t=new char[length];
cout<<"Character array of length"<<length<<"created";
}
~sample ()
{
delete t;
cout<<"\nMemory de-allocated for the character array";
}
};
int main()
{
sample s (10);
return 0;
}
```

प्रोग्राम का आउटपुट होगा
Character array of length 10 created
Memory de-allocated for the character array

निबंधात्मक प्रश्न

प्रश्न 1. पैरामीटराइज्ड कंस्ट्रक्टर की सहायता से एक आयत (rectangle) का क्षेत्रफल (area) ज्ञात करने का प्रोग्राम (program) लिखिए।

उत्तर-

```
#include<iostream>
using namespace std;
class rectangle
{
int length;
int breadth;
public:
rectangle (int a, int b)
{
length=a;
breadth=b;
}
void area ()
{
cout<<"Area="<<length*breadth;
}
};
int main ()
{
rectangle r (5,10);
return 0;
}
प्रोग्राम का आउटपुट होगा
Area= 50
```

प्रश्न 2. कंस्ट्रक्टर ओवरलोडिंग से आप क्या समझते हैं? एक प्रोग्राम द्वारा स्पष्ट कीजिए।

अथवा

बिना आरग्यूमेन्ट का कंस्ट्रक्टर (no argument constructor), एक आरग्यूमेन्ट का कंस्ट्रक्टर (one argument constructor) और दो आरग्यूमेन्ट का कंस्ट्रक्टर (two argument constructor) को एक प्रोग्राम द्वारा समझाइए।

उत्तर- कंस्ट्रक्टर ओवरलोडिंगएक – क्लास में एक से ज्यादा कंस्ट्रक्टर हो सकते हैं और इसे कंस्ट्रक्टर ओवरलोडिंग कहा जाता है।

प्रोग्राम-कंस्ट्रक्टर ओवरलोडिंग

```
#include<iostream>
using namespace std;
class point
{
int x,y;
public:
point () // no argument constructor
{
x=0;
y=0;
}
point (int a) // one argument constructor
{x=y=a; }
point (int m, int n)//two arguments constructor
{
x=m;
y=n;
}
void show()
{
cout<<"x="<<x<<"\n";
cout<<"y="<<y<<"\n";
}
};
int main()
{
point p1;
point p2 (5);
point p3 (7,11);
cout<<"Coordinates of p1 are\n";
p1. show ();
cout<<"Coordinates of p2 are\n";
p2 . show ();
cout<<"Coordinates of p3 are\n";
p3. show ();
```

```

return 0;
}
प्रोग्राम का आउटपुट होगा
Coordinates of p1 are
x=0
y=0
Coordinates of p2 are
x=5
y=5
Coordinates of p3 are
x=7
y=11

```

उपरोक्त प्रोग्राम में, क्लास point में तीन कंस्ट्रक्टर हैं। पहला बिना आरग्यूमेन्ट का कंस्ट्रक्टर है। इसका उपयोग बिना किसी प्रारंभिक वेल्डू का ऑब्जेक्ट बनाने के लिए किया जाता है। जब हम अन्य कंस्ट्रक्टर परिभाषित करते हैं तब हमें कम्पाइलर को संतुष्ट करने के लिए बिना आरग्यूमेन्ट का कंस्ट्रक्टर परिभाषित करना होगा। दूसरा कंस्ट्रक्टर एक वेल्डू पैरामीटर के रूप में लेता है और ऑब्जेक्ट को इनिशियलाइज करता है। तीसरा कंस्ट्रक्टर दो आरग्यूमेन्ट लेता है और इन दो वेल्डू से ऑब्जेक्ट को इनिशियलाइज करता है।

प्रश्न 3. ऑब्जेक्ट के डाइनामिक इनिशियलाइजेशन को एक प्रोग्राम द्वारा समझाइए।

अथवा

किसी वृत्त का तथा आयत का क्षेत्रफल निकालने का प्रोग्राम डाइनामिक इनिशियलाइजेशन की सहायता से लिखिए।

उत्तर- ऑब्जेक्ट का डाइनामिक इनिशियलाइजेशन-एक ऑब्जेक्ट की इनिशियल वेल्डू रन टाइम पर दी जा सकती है। डाइनामिक इनिशियलाइजेशन का फायदा यह है कि हम विभिन्न प्रकार के इनपुट कंस्ट्रक्टर ओवरलोडिंग के द्वारा दे सकते हैं।

```

#include<iostream>
using namespace std;
class shape
float length, breadth;
float radius;
float area;
public:
shape () {}

```

```

shape (float r)
{
radius=r;
area=3.14* r* r;
}
shape (float l, float b)
{
length=l;
breadth=b;
area=length*breadth;
}
void display ( )
{
cout<<"Area="<<area<<"\n";
}
};
int main()
{
shape circle, rectangle;
float r, l, b;
cout<<"Enter the radius of circle\n";
cin>>r;
circle=shape (r);
cout<<"Enter the length and breadth of rectangle\n";
cin>>l>>b;
rectangle=shape (l,b);
cout<<"Area of circle\n";
circle.display();
cout<<"Area of rectangle\n";
rectangle.display();
return 0;
}

```

प्रोग्राम का आउटपुट होगा

Enter the radius of circle

5

Enter the length and breadth of rectangle

17 8

Area of circle

78.5

Area of rectangle

136

प्रश्न 4. कॉपी कंस्ट्रक्टर को उदाहरण सहित समझाइए।

उत्तर- कॉपी कंस्ट्रक्टर-कंस्ट्रक्टर जो एक ऑब्जेक्ट को उसी क्लास के दूसरे ऑब्जेक्ट से घोषित और इनिशियलाइज करता है, उसे कॉपी कंस्ट्रक्टर कहा जाता है। एक कॉपी कंस्ट्रक्टर उसी क्लास के ऑब्जेक्ट का रेफरेंस आरग्यूमेन्ट के रूप में लेते हैं।

प्रोग्राम-कॉपी कंस्ट्रक्टर

```
#include <iostream>
using namespace std;
class product
{
int code;
public:
product () { } //default constructor
product (int x) //parameterized constructor
{
code=x;
}
product (product &y) //copy constructor
{
code=y.code; //copy the value
}
void display(void)
{
cout<<code;
}
int main()
{
product p1(10);
product p2 (p1); //copy constructor called
product p3=p1; //again copy constructor called
cout<<"Code of p1:";
p1.display();
cout<<"\n Code of p2:";
p2.display();
cout<<"\n Code of p3:";
```

```
p3.display();
```

```
return 0;
```

```
}
```

प्रोग्राम का आउटपुट होगा

Code of p1:10

Code of p2:10

Code of p3:10