

ऑपरेटर, एक्सप्रेशन और कंट्रोल स्ट्रक्चर

पाठ्यपुस्तक के प्रश्न

वस्तुनिष्ठ प्रश्न

प्रश्न 1. इनमें से कौनसा ऑपरेटर इसके दायीं तरफ के वेरिएबल के कन्टेन्ट को आउटपुट स्क्रीन पर प्रिन्ट करता है?

- (अ) < <
- (ब) > >
- (स) ::
- (द) &

उत्तर: (द) &

प्रश्न 2. इनमें से कौनसा ऑपरेटर पर्याप्त मात्रा में डाटा ऑब्जेक्ट को मैमोरी प्रदान करता है?

- (अ) Insertion ऑपरेटर
- (ब) delete ऑपरेटर
- (स) new ऑपरेटर
- (द) delete ऑपरेटर

उत्तर: (स) new ऑपरेटर

प्रश्न 3. एक्सप्रेशन $a = (b = 20) + 5$ में वेरिएबल 'a' का मान होगा?

- (अ) 20
- (ब) 25
- (स) 5
- (द) 30

उत्तर: (ब) 25

प्रश्न 4. इनमें से कौनसा शॉर्ट हैंड असाइनमेन्ट ऑपरेटर है?

- (अ) +=
- (ब) -=
- (स) *=
- (द) ये सभी

उत्तर: (द) ये सभी

प्रश्न 5. सलेक्शन स्ट्रक्चर को किस कन्ट्रोल स्टेटमेंट के द्वारा लागू किया गया है?

- (अ) if स्टेटमेंट
- (ब) if-else स्टेटमेंट
- (स) switch स्टेटमेंट
- (द) ये सभी

उत्तर: (द) ये सभी

प्रश्न 6. लूप स्ट्रक्चर को किस कन्ट्रोल स्टेटमेंट के द्वारा लागू किया गया है?

- (अ) for स्टेटमेंट
- (ब) while स्टेटमेंट
- (स) do-while स्टेटमेंट
- (द) ये सभी

उत्तर: (द) ये सभी

अतिलघु उत्तरीय प्रश्न

प्रश्न 1. ऑपरेटर प्रिसीडेंस को परिभाषित करें?

उत्तर- ऑपरेटर प्रिसीडेंस-अगर एक से ज्यादा ऑपरेटर किसी एक्सप्रेसन में हो तब, C++ भाषा में ऑपरेटर की प्राथमिकता के लिए परिभाषित नियम होते हैं। उच्च प्राथमिकता वाले ऑपरेटर निम्न प्राथमिकता वाले ऑपरेटर से पहले संपादित होते हैं। इस नियम को ऑपरेटर प्रिसीडेंस कहा जाता है।

प्रश्न 2. ऑपरेटर की सम्बद्धता को परिभाषित करें।

उत्तर- ऑपरेटर की सम्बद्धता (एसोसिएटिविटी)-अगर दो या दो से अधिक ऑपरेटर एक समान प्रिसीडेंस के एक ही एक्सप्रेसन में होते हैं तो जिस आर्डर में वे संपादित होते हैं उसे ऑपरेटर की सम्बद्धता (एसोसिएटिविटी) कहते हैं।

प्रश्न 3. विभिन्न प्रकार के कन्ट्रोल स्ट्रक्चर क्या होते हैं?

उत्तर- कन्ट्रोल स्ट्रक्चर तीन प्रकार के होते हैं

- सिक्केस स्ट्रक्चर
- सलेक्शन स्ट्रक्चर

- लूप स्ट्रक्चर

प्रश्न 4. एक्सप्रेशनस क्या होते हैं?

उत्तर- एक एक्सप्रेशनस ऑपरेटर, कांस्टेंट और वेरिएबल का कॉम्बिनेशन है जो भाषा के नियम के अनुसार व्यवस्थित होता है।

लघु उत्तरीय प्रश्न

प्रश्न 1. स्कोप रिजोल्यूशन ऑपरेटर का क्या उपयोग है?

उत्तर: स्कोप रिजोल्यूशन ऑपरेटर (::)-C++ एक ब्लॉक स्ट्रक्चर भाषा है। एक वेरिएबल के नाम को अलग-अलग ब्लॉक में प्रयोग कर सकते हैं। वेरिएबल का स्कोप इसकी घोषणा की जगह और ब्लॉक के अंत के बीच में होता है। एक वेरिएबल जिसकी घोषणा ब्लॉक के अन्दर की गयी है वह उस ब्लॉक के लिए लोकल होता है। स्कोप रिजोल्यूशन ऑपरेटर का प्रयोग वेरिएबल के ग्लोबल वर्जन को एक्सेस करने के लिए किया जाता है।

उदाहरण: प्रोग्राम-स्कोप रिजोल्यूशन ऑपरेटर

```
#include
using namespace std;
int x=10; //global variable
int main()
{
int x =20; //x re-declared, local to main
{
cout<<"Inner block \n";
int x = 30; //x declared again, local to inner block
cout<<"x="<<x<<"\n";
cout<<"::x="<<::x<<"\n";
}
cout<<"Outer block\n";
cout<<"x="<<x<<"\n";
return 0;
}
प्रोग्राम का आउटपुट होगा
Inner block
x= 30
:: x = 10
```

Outer block

x = 20

:: x = 10

प्रश्न 2. new और delete ऑपरेटर का क्या उपयोग है?

उत्तर:

- new ऑपरेटर – यह ऑपरेटर पर्याप्त मात्रा में डाटा ऑब्जेक्ट को मेमोरी प्रदान करता है।
int *p=new int;
उपरोक्त स्टेटमेंट इंटीजर डाटा ऑब्जेक्ट को पर्याप्त मात्रा में मेमोरी प्रदान करता है।
- delete ऑपरेटर – यह ऑपरेटर मेमोरी को पुनः आवंटित करता है। जब डेटा ऑब्जेक्ट की आगे आवश्यकता नहीं होती है। इससे मुक्त की गयी मेमोरी दूसरी प्रोग्रामों के लिए पुनः उपयोग में लायी जा सकती है।
उदाहरण
delete p;
उपरोक्त स्टेटमेंट मेमोरी जिसे पॉइन्टर p द्वारा अंकित किया गया है उसे पुनः आवंटित करता है।

प्रश्न 3. C++ में सलेक्शन कंट्रोल स्ट्रक्चर कैसे लागू किया गया है? वर्णन कीजिए।

उत्तर: सलेक्शन स्ट्रक्चर-दो या उससे अधिक संपादन के पथ जिनमें से एक को चुना जाता है अगर शर्त पूरी होती है।

उदाहरण: if स्टेटमेन्ट

```
if (expression is true)
```

```
{
```

```
statements;
```

```
}
```

if-else स्टेटमेन्ट

```
if (expression is true)
```

```
{
```

```
statements;
```

```
}
```

```
else
```

```
{
```

```
statements;
```

```
}
```

switch स्टेटमेन्ट

```
switch(expression)
{
case 1 : statements;
break;
case 2: statements;
break;
case 3 : statements;
break;
default : statements;
}
```

निबंधात्मक प्रश्न

प्रश्न 1. विभिन्न प्रकार के एक्सप्रेशन का उदाहरण सहित वर्णन कीजिए।

उत्तर- एक एक्सप्रेशन ऑपरेटर, कांस्टेंट और वेरिएबल का कॉम्बिनेशन है जो भाषा के नियम के अनुसार व्यवस्थित होता है। एक्सप्रेशन के निम्न प्रकार होते हैं

- कांस्टेंट एक्सप्रेशन
- इंटीग्रल एक्सप्रेशन
- फ्लोट एक्सप्रेशन
- पॉइन्टर एक्सप्रेशन
- रिलेशनल एक्सप्रेशन
- लोजिकल एक्सप्रेशन
- बिटवाइज एक्सप्रेशन

कांस्टेंट एक्सप्रेशन – इसमें केवल कांस्टेंट वैल्यूज होती है।

उदाहरण- $20+10*5.2$

इंटीग्रल एक्सप्रेशन – जो एक्सप्रेशन स्वतः और बाह्य टाईप कनवर्जन के बाद इंटीजर परिणाम देते हैं।

उदाहरण- $x+y*10$

$x+'a'$

$5+\text{int}(7.5)$

जहाँ x और y इंटीजर वेरिएबल है।

फ्लोट एक्सप्रेशन – जो एक्सप्रेशन सभी तरह के टाईप कनवर्जन के बाद फ्लोट टाईप परिणाम देते हैं।

उदाहरण – $a+b/5$

$7+\text{float}(10)$

जहाँ a और b फ्लोट टाईप के वेरिएबल हैं।

पॉइन्टर एक्सप्रेशन – पॉइन्टर का परिणाम एड्रेस वेन्यू होता है।

उदाहरण – $\text{ptr} = \&x;$

$\text{ptr}++$

जहाँ x एक वेरिएबल है और ptr एक पॉइन्टर है।

रिलेशनल एक्सप्रेशन – जो एक्सप्रेशन बूलियन टाईप का परिणाम देते हैं। जो सत्य और असत्य हो सकता है।

उदाहरण – $x \leq y$

$a == b$

रिलेशनल एक्सप्रेशन को बूलियन एक्सप्रेशन भी कहा जाता है।

लोजिकल एक्सप्रेशन – जो एक्सप्रेशन दो या दो से अधिक रिलेशनल एक्सप्रेशन को जोड़ता है और बूलियन टाईप का परिणाम देते हैं।

उदाहरण – $x > y \ \&\& \ x == 5$

$a = 20 \ || \ y = 10$

बिटवाइज एक्सप्रेशन – इस तरह के एक्सप्रेशन को बिट स्तर के डेटा स्तर के डेटा मैन्युपूलेशन के लिए प्रयोग किया जाता है। इनका उपयोग बिट्स की टेस्टिंग और शिफ्टिंग के लिए किया जाता है।

उदाहरण

$a < < 3$ // तीन बिट्स को बायीं तरफ शिफ्ट करता है।

$x > > 1$ // एक बिट को दायीं तरफ शिफ्ट करता है।

प्रश्न 2. विभिन्न प्रकार के लूपिंग स्टेटमेंटों का वर्णन कीजिए।

उत्तर- for स्टेटमेंट-for लूप का प्रयोग किया जाता है जब किसी कार्य को पूर्व निर्धारित संख्या के बराबर दोहराया जाता है।

`for(initial value;test condition;increment/decrements)`

```
{  
statements;  
}
```

`while` स्टेटमेंट

`while` लूप के अन्दर के स्टेटमेंट जब तक कंडिशन सत्य है तब तक संपादित होता है। इसे प्रि-टेस्ट कंडिशन लूप भी कहा जाता है।

```
while(condition is true)
```

```
{
```

```
statements;
```

```
}
```

```
do-while स्टेटमेंट
```

do-while लूप को पोस्ट टेस्ट कंडिशन लूप कहा जाता है यह लूप कम से कम एक बार तो संपादित होता है।

```
do
```

```
{
```

```
statements;
```

```
}
```

```
while(condition is true);
```

अन्य महत्वपूर्ण प्रश्न

अतिलघु उत्तरीय प्रश्न

प्रश्न 1. C++ में कुछ नये ऑपरेटर के नाम बताइए।

उत्तर- Insertion ऑपरेटर (<<), Extraction ऑपरेटर (>>), स्कोप रिजोल्यूशन ऑपरेटर (::)

प्रश्न 2. Insertion ऑपरेटर का कार्य बताइए।

उत्तर- Insertion ऑपरेटर (<<) अपने दायीं तरफ के वेरिएबल के कन्टेन्ट को आउटपुट स्क्रीन पर प्रिन्ट करता है।

प्रश्न 3. Extraction ऑपरेटर का कार्य बताइए।

उत्तर- Extraction ऑपरेटर (>>) कीबोर्ड से वैल्यू लेता है और इसके दायीं तरफ के वेरिएबल को प्रदान करता है।

प्रश्न 4. स्कोप रिजोल्यूशन ऑपरेटर का प्रयोग कब किया जाता है?

उत्तर- स्कोप रिजोल्यूशन ऑपरेटर का प्रयोग वेरिएबल के ग्लोबल वर्जन को एक्सेस करने के लिए किया जाता है।

प्रश्न 5. new ऑपरेटर क्या करता है?

उत्तर- new ऑपरेटर पर्याप्त मात्रा में डाटा ऑब्जेक्ट को मैमोरी प्रदान करता है।

प्रश्न 6. इंटीग्रल एक्सप्रेशन के बारे में बताइए।

उत्तर- जो एक्सप्रेशन स्वतः और बाय टाईप कनवर्जन के बाद इंटीजर परिणाम देते हैं। इंटीग्रल एक्सप्रेशन कहलाते हैं।

प्रश्न 7. फ्लोट एक्सप्रेशन क्या होते हैं?

उत्तर- जो एक्सप्रेशन सभी तरह के टाईप कनवर्जन के बाद फ्लोट टाईप परिणाम देते हैं। फ्लोट एक्सप्रेशन कहलाते हैं।

प्रश्न 8. बिटवाइज एक्सप्रेशन का उपयोग क्या होता है?

उत्तर- बिटवाइज एक्सप्रेशन का उपयोग बिट्स की टेस्टिंग और शिफ्टिंग के लिए किया जाता है।

प्रश्न 9. for लूप का प्रयोग कब किया जाता है?

उत्तर- जब किसी कार्य को पूर्व निर्धारित संख्या के बराबर दोहराया जाता है। तब for लूप का प्रयोग किया जाता है।

प्रश्न 10. for लूप का syntax लिखिए।

उत्तर- for(initial value;test condition ;increments/decrements)
{
statements;
}

प्रश्न 11. While लूप का दूसरा नाम बताइए।

उत्तर- while लूप को प्रि-टेस्ट कंडीशन लूप भी कहा जाता है।

प्रश्न 12. While लूप का Syntax लिखिए।

उत्तर- while (condition is true)
{


```
statements;  
}
```

प्रश्न 13. do-while लूप का दूसरा नाम बताइए।

उत्तर- do-while लूप को पोस्ट टेस्ट कंडीशन लूप कहा जाता है।

प्रश्न 14. do-while लूप का Syntax बताइए।

उत्तर-

```
do  
{  
statements;  
}  
while (condition is true);
```

लघु उत्तरीय प्रश्न

प्रश्न 1. ऑपरेटर की एसोसिएटिविटी को विस्तार से समझाइए।

उत्तर- ऑपरेटर की एसोसिएटिविटी-अगर दो या दो से अधिक ऑपरेटर एक समान प्रिसिडेंस के एक ही एक्सप्रेसन में होते हैं तो जिस आर्डर में वे संपादित होते हैं उसे ऑपरेटर की एसोसिएटिविटी कहते हैं। C++ ऑपरेटर की सम्पूर्ण लिस्ट उनकी प्रिसिडेंस और एसोसिएटिविटी के साथ नीचे टेबल में दी गयी है। टेबल ऑपरेटर प्रिसिडेंस और एसोसिएटिविटी

ऑपरेटर प्रिसिडेंस	एसोसिएटिविटी
::	बायें से दायें
>, ., (), [], ++, --, ~, !, unary+, unary-, unary*	बायें से दायें
Unary &, (type), sizeof, new, delete	दायें से बायें
*, /, %	बायें से दायें
+, -	बायें से दायें
<<, >>	बायें से दायें
<, <=, >, >=	बायें से दायें
=, !=	बायें से दायें

&	बायें से दायें
^	बायें से दायें
	बायें से दायें
&&	बायें से दायें
	बायें से दायें
?:	बायें से दायें
=, *=, /=, %=, +=	दायें से बायें
<<=, >>=, &=, ^=, =, (comma)	बायें से दायें

प्रश्न 2. विशेष असाइनमेन्ट एक्सप्रेसनस के विषय में बताइए।

उत्तर- विशेष असाइनमेन्ट एक्सप्रेसनस

- चैनड असाइनमेन्ट – $a=b=10$;
पहले 10 वैल्यू b को प्रदान की जाती है उसके बाद a को।
- एम्बेडेड असाइनमेन्ट – $a=(b=20)+5$;
($b=20$) एक असाइनमेन्ट एक्सप्रेसन है जिस एम्बेडेड असाइनमेन्ट कहा जाता है। यहाँ पर वैल्यू 20, b को दी जाती है और उसके बाद परिणाम 25, a को दिया जाता है।
- कम्पाउंड असाइनमेन्ट – यह असाइनमेन्ट ऑपरेटर और एक बाइनरी अर्थमैटिक ऑपरेटर का संयुक्त रूप है।
उदाहरण – $a=a+5$; को $a+=5$; के रूप में लिख सकते हैं।
 $+=$ ऑपरेटर को कम्पाउंड असाइनमेन्ट ऑपरेटर या शॉर्ट हेड असाइनमेन्ट ऑपरेटर कहा जाता है।