|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which of the following escape sequence represents carriage return? |
| ((OPTION\_A)) | \r |
| ((OPTION\_B)) | \c |
| ((OPTION\_C)) | \n\r |
| ((OPTION\_D)) | None of the above |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 2 |
| ((QUESTION)) | Input: 1  #include <stdio.h>  int main()  {  register int i, x;  scanf("%d", &i);  x = ++i;  printf("%d", x);  return 0;  } |
| ((OPTION\_A)) | 0 |
| ((OPTION\_B)) | 1 |
| ((OPTION\_C)) | 2 |
| ((OPTION\_D)) | Compiler Error |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 2 |
| ((QUESTION)) | #include <iostream>  using namespace std;  #include <iostream>  using namespace std;  int main() {  int i, x[5], y, z[5];  for (i = 0; i < 5; i++) {  x[i] = i;  z[i] = i + 3;  y = z[i];  x[i] = y++;  }  for (i = 0; i < 5; i++)  cout << x[i] << " ";  return 0;  } |
| ((OPTION\_A)) | 3 4 5 6 7 |
| ((OPTION\_B)) | 4 5 6 7 8 |
| ((OPTION\_C)) | 2 3 4 5 6 |
| ((OPTION\_D)) | none of above |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | #include <stdio.h>  #define square(x) (x \* x)  int main()  {  int x, y = 3;  x = square(y + 1);  printf("%d\n", x);  return 0;  } |
| ((OPTION\_A)) | Error |
| ((OPTION\_B)) | 16 |
| ((OPTION\_C)) | 7 |
| ((OPTION\_D)) | Garbage |
| ((CORRECT\_CHOICE)) (A/B/C/D) |  |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | #include <iostream>  using namespace std;  int main()  {  char\* str = "HelloWorld";  int\* p, arr[] = { 10, 15, 70, 19 };  p = arr;  str++;  p++;  cout << \*p << " " << str << endl;  return 0;  } |
| ((OPTION\_A)) | 10 elloWorld |
| ((OPTION\_B)) | 15 HelloWorld |
| ((OPTION\_C)) | 15 elloWorld |
| ((OPTION\_D)) | 10 HelloWorld |
| ((CORRECT\_CHOICE)) (A/B/C/D) | C |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | #include <stdio.h>  void main()  {  unsigned char x = 350;  printf("%d", x);  } |
| ((OPTION\_A)) | Error |
| ((OPTION\_B)) | 94 |
| ((OPTION\_C)) | 350 |
| ((OPTION\_D)) | Garbage Value |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 2 |
| ((QUESTION)) | #include <stdio.h>  int main(void)  {  int y, z;  int x = scanf("%d %d", &y, &z);  printf("%d", x);  return 0;  }  Input: 12 10 |
| ((OPTION\_A)) | 12 |
| ((OPTION\_B)) | 2 |
| ((OPTION\_C)) | Syntax error |
| ((OPTION\_D)) | 10 |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | void main()  {  int const\* p = 5;  printf("%d", ++(\*p));  } |
| ((OPTION\_A)) | 6 |
| ((OPTION\_B)) | 5 |
| ((OPTION\_C)) | Garbage Value |
| ((OPTION\_D)) | Compiler Error |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | What is the general syntax for accessing the namespace variable? |
| ((OPTION\_A)) | namespace::operator |
| ((OPTION\_B)) | namespace:operator |
| ((OPTION\_C)) | namespace->operator |
| ((OPTION\_D)) | none of the mentioned |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | A constructor that accept \_\_\_\_\_ parameters is called the default constructor. |
| ((OPTION\_A)) | 0 |
| ((OPTION\_B)) | 1 |
| ((OPTION\_C)) | 2 |
| ((OPTION\_D)) | 3 |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | #include <iostream>  using namespace std;  int main()  {  for (int i = 14; i > 8; i = i - 2)  cout << i;  for (int i = -5; i > -7; i--)  cout << i + 1;  return 0;  } |
| ((OPTION\_A)) | 141210-4-5 |
| ((OPTION\_B)) | 141210-4 |
| ((OPTION\_C)) | 1412-4-5 |
| ((OPTION\_D)) | 141210-4 |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | #include"stdio.h"  #include"stdlib.h"  void reverse(int i)  {  if (i > 5)  exit(0);  printf("%d\n", i);  return reverse(i++);  }  int main()  {  reverse(1);  } |
| ((OPTION\_A)) | Segmenation fault |
| ((OPTION\_B)) | Compilation error |
| ((OPTION\_C)) | Print 1 Infinite time |
| ((OPTION\_D)) | Both a & c |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | The number of relational operators in the C language is |
| ((OPTION\_A)) | 4 |
| ((OPTION\_B)) | 6 |
| ((OPTION\_C)) | 3 |
| ((OPTION\_D)) | 1 |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | # include <stdio.h>  void fun(int \*ptr)  {  \*ptr = 30;  }    int main()  {  int y = 20;  fun(&y);  printf("%d", y);    return 0;  } |
| ((OPTION\_A)) | 20 |
| ((OPTION\_B)) | 30 |
| ((OPTION\_C)) | Compile Error |
| ((OPTION\_D)) | Runtime Error |
| ((CORRECT\_CHOICE)) (A/B/C/D) |  |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | In CPP, cin and cout are the predefined stream \_\_\_\_\_\_\_\_\_\_ . |
| ((OPTION\_A)) | Operator |
| ((OPTION\_B)) | Functions |
| ((OPTION\_C)) | Objects |
| ((OPTION\_D)) | Data types |
| ((CORRECT\_CHOICE)) (A/B/C/D) | C |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | #include <stdio.h>  int main()  {  int \*\*\*r, \*\*q, \*p, i=8;  p = &i;  q = &p;  r = &q;  printf(“%d, %d, %d\n”, \*p, \*\*q, \*\*\*r);  return 0;  } |
| ((OPTION\_A)) | 8, 8, 8 |
| ((OPTION\_B)) | 4000, 4002, 4004 |
| ((OPTION\_C)) | 4000, 4004, 4008 |
| ((OPTION\_D)) | 4000,4000, 4000 |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/`/3...) | 1 |
| ((QUESTION)) | #include <stdio.h>  int main()  {  int i=5, \*j, k;  j = &i;  printf(“%d\n”, i\*\*j\*i+\*j);  return 0;  } |
| ((OPTION\_A)) | 130 |
| ((OPTION\_B)) | 625 |
| ((OPTION\_C)) | 25 |
| ((OPTION\_D)) | 20 |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | #include <iostream>  using namespace std;  int main()  {  int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};  cout << \*(a[1] + 3) << \*(\*(a + 1) + 3) << 3[1[a]];  return 0;  } |
| ((OPTION\_A)) | 242424 |
| ((OPTION\_B)) | 182124 |
| ((OPTION\_C)) | 212121 |
| ((OPTION\_D)) | None of the above |
| ((CORRECT\_CHOICE)) (A/B/C/D) | A |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | Which of the following correctly shows the hierarchy of arithmetic operations in C? |
| ((OPTION\_A)) | / + \* - |
| ((OPTION\_B)) | \* - / + |
| ((OPTION\_C)) | + - / \* |
| ((OPTION\_D)) | / \* + - |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 2 |
| ((QUESTION)) | #include <iostream>  using namespace std;  class Test {  static int i;  int j,k;  }  int Test::i;  int main() {  cout<<sizeof(Test);  return 0;  }  Assume size of int as 4 bytes |
| ((OPTION\_A)) | 12 |
| ((OPTION\_B)) | 8 |
| ((OPTION\_C)) | 4 |
| ((OPTION\_D)) | Error |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | What should be the output of below program?  #include <stdio.h>  int main()  {  if(printf("hello ")){  printf("world");  }  } |
| ((OPTION\_A)) | world |
| ((OPTION\_B)) | hello world |
| ((OPTION\_C)) | hello |
| ((OPTION\_D)) | Compilation error |
| ((CORRECT\_CHOICE)) (A/B/C/D) |  |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | #include <iostream>  using namespace std;  int main()  {  int i;  i = 1 + (1,4,5,6,3);  cout << i;  return 0;  } |
| ((OPTION\_A)) | 2 |
| ((OPTION\_B)) | 5 |
| ((OPTION\_C)) | 1 |
| ((OPTION\_D)) | 4 |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 2 |
| ((QUESTION)) | #include <stdio.h>  void main()  {  printf(6 + "Hello World");  } |
| ((OPTION\_A)) | 6Hello World |
| ((OPTION\_B)) | Hello World |
| ((OPTION\_C)) | World |
| ((OPTION\_D)) | Hello |
| ((CORRECT\_CHOICE)) (A/B/C/D) | C |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | void main()  {  int a = 1;  switch(a)  {  case 1: cout<<"1";  case 2: cout<<"2";  case 3: cout<<"3";  default: cout<<"Default";  }  } |
| ((OPTION\_A)) | 1 |
| ((OPTION\_B)) | Default |
| ((OPTION\_C)) | Error |
| ((OPTION\_D)) | 123 |
| ((CORRECT\_CHOICE)) (A/B/C/D) | D |
| ((EXPLANATION)) (OPTIONAL) |  |

|  |  |
| --- | --- |
| ((MARKS)) (1/2/3...) | 1 |
| ((QUESTION)) | int main()  {  int a[10];  printf("%d",\*a+1-\*a+3);  return 0;  } |
| ((OPTION\_A)) | Garbage |
| ((OPTION\_B)) | 4 |
| ((OPTION\_C)) | 2 |
| ((OPTION\_D)) | Error |
| ((CORRECT\_CHOICE)) (A/B/C/D) | B |
| ((EXPLANATION)) (OPTIONAL) |  |